

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-190993

(43)Date of publication of application : 13.07.1999

(51)Int.Cl.

G10H 1/00

G10H 1/00

G10K 15/04

(21)Application number : 10-058438

(71)Applicant : YAMAHA CORP

(22)Date of filing : 10.03.1998

(72)Inventor : KIKUCHI TAKESHI

(30)Priority

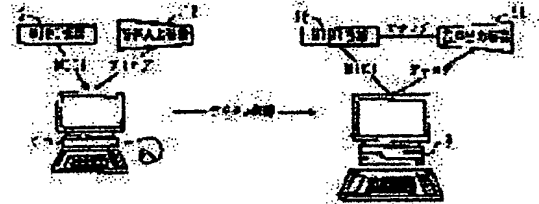
Priority number : 09290093 Priority date : 22.10.1997 Priority country : JP

(54) DEVICE AND METHOD FOR COMMUNICATING MUSICAL SOUND INFORMATION, DEVICE AND METHOD FOR CONTROL, AND MEDIUM WHERE PROGRAM IS RECORDED

(57)Abstract

PROBLEM TO BE SOLVED: To reproduce a plurality of pieces of musical sound information synchronously by communicating them.

SOLUTION: An imparting means (encoder) 3 imparts time information on a common time base to each of 1st and 2nd pieces of musical sound information, and a transmitting means (encoder) 3 relates the time information given to the 1st and 2nd pieces of musical sound information to the respective pieces of musical sound information and transmits it, and a receiving means 9 (home computer) which receives the 1st and 2nd pieces of musical sound information and time information related to the pieces of musical sound information from the transmitting means 3 and an output means (home computer) 9 synchronizes the 1st and 2nd pieces of musical sound information on the basis of the time information and outputs them to a reproducing device.



LEGAL STATUS

[Date of request for examination] 08.09.1999

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 3196715

[Date of registration] 08.06.2001

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2000 Japan Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-190993

(43) 公開日 平成11年(1999) 7月13日

(51) Int.Cl.⁵
G 1 0 H 1/00
G 1 0 K 15/04

識別記号
1 0 2
3 0 2

F I
G 1 0 H 1/00 Z
1 0 2 Z
G 1 0 K 15/04 3 0 2 D

審査請求 未請求 請求項の数78 O L (全 23 頁)

(21) 出願番号 特願平10-58438
(22) 出願日 平成10年(1998) 3月10日
(31) 優先権主張番号 特願平9-290093
(32) 優先日 平 9 (1997) 10月22日
(33) 優先権主張国 日本 (J P)

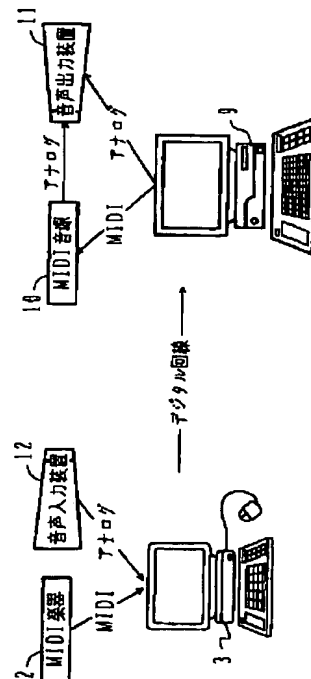
(71) 出願人 000004075
ヤマハ株式会社
静岡県浜松市中沢町10番1号
(72) 発明者 菊池 健
静岡県浜松市中沢町10番1号 ヤマハ株式
会社内
(74) 代理人 弁理士 高橋 敬四郎 (外1名)

(54) 【発明の名称】 楽音情報の通信装置、通信方法、制御装置、制御方法及びプログラムを記録した媒体

(57) 【要約】

【課題】 複数の楽音情報を通信して同期をとりながら再生するための楽音情報の処理装置、処理方法又はプログラムを記録した媒体を提供することを課題とする。

【解決手段】 第1及び第2の楽音情報の各々に、共通の時間軸上の時間情報を付与する付与手段(3)と、第1及び第2の楽音情報の各々と、各楽音情報に付与された時間情報とを関連付けて送信する送信手段(3)と、送信手段により送信されたものであって第1及び第2の楽音情報及び各楽音情報に関連付けられた時間情報を受信する受信手段(9)と、時間情報を基に前記第1及び第2の楽音情報の同期をとって再生装置に出力する出力手段(9)とを有する。



【特許請求の範囲】

【請求項1】 第1及び第2の楽音情報の各々に、共通の時間軸上の時間情報を付与する付与手段と、前記第1及び第2の楽音情報の各々と、各楽音情報に付与された時間情報とを関連付けて送信する送信手段とを有する楽音情報の通信装置。

【請求項2】 前記第1及び第2の楽音情報は異なる楽音情報生成装置で生成された楽音情報である請求項1記載の楽音情報の通信装置。

【請求項3】 前記第1の楽音情報はMIDIデータであり、前記第2の楽音情報は音声データである請求項2記載の楽音情報の通信装置。

【請求項4】 前記付与手段は、第1及び第2の楽音情報の各々を楽音パケットに分割して各楽音パケットに時間情報を付与して送信パケットを生成し、前記送信手段は、前記送信パケットを送信する請求項1～3のいずれかに記載の楽音情報の通信装置。

【請求項5】 前記送信手段はインターネット上に前記送信パケットを送信する請求項4記載の楽音情報の通信装置。

【請求項6】 さらに、演奏者の演奏操作に応じてリアルタイムで生成されたMIDIデータ、及び音声リアルタイムで電気信号に変換された音声データを入力する入力手段を有し、前記送信手段は、前記入力されたMIDIデータ及び音声データをリアルタイムで送信する請求項3記載の楽音情報の通信装置。

【請求項7】 さらに、前記送信手段により送信されたものであって前記第1及び第2の楽音情報及び各楽音情報に関連付けられた時間情報を受信する受信手段と、前記時間情報を基に前記第1及び第2の楽音情報の同期をとって再生装置に出力する出力手段とを有する請求項1～6のいずれかに記載の楽音情報の通信装置。

【請求項8】 前記出力手段は、前記受信した第1及び第2の楽音情報をリアルタイムで再生装置に出力する請求項7記載の楽音情報の通信装置。

【請求項9】 前記出力手段は定期的に第1及び第2の楽音情報の同期をとる請求項7又は8記載の楽音情報の通信装置。

【請求項10】 前記出力手段は、第1及び第2の楽音情報の間の時間的ずれの有無を定期的に検出し、時間的ずれを検出したときには該時間的ずれを修正する請求項9記載の楽音情報の通信装置。

【請求項11】 前記第1又は第2の楽音情報は音声データであり、前記出力手段は、前記第1又は第2の楽音情報の間引き又は補間を行うことにより前記時間的ずれを修正する請求項10記載の楽音情報の通信装置。

【請求項12】 前記第1又は第2の楽音情報は複数のMIDIイベントと該複数のMIDIイベント間の時間間隔情報を含み、前記出力手段は、前記時間間隔情報を

調整することにより前記時間的ずれを修正する請求項10記載の楽音情報の通信装置。

【請求項13】 第1及び第2の楽音情報及び各楽音情報に関連付けられた時間情報を受信する受信手段と、前記時間情報を基に前記第1及び第2の楽音情報の同期をとって再生装置に出力する出力手段とを有する楽音情報の通信装置。

【請求項14】 前記第1及び第2の楽音情報は異なる楽音情報生成装置により生成された楽音情報である請求項13記載の楽音情報の通信装置。

【請求項15】 前記第1の楽音情報はMIDIデータであり、前記第2の楽音情報は音声データである請求項14記載の楽音情報の通信装置。

【請求項16】 前記受信手段は、第1及び第2の楽音情報の各々が楽音パケットに分割され、各楽音パケットに時間情報が付与された送信パケットを受信する請求項13～15のいずれかに記載の楽音情報の通信装置。

【請求項17】 前記受信手段はインターネット上から前記送信パケットを受信する請求項16記載の楽音情報の通信装置。

【請求項18】 前記出力手段は、前記受信した第1及び第2の楽音情報をリアルタイムで再生装置に出力する請求項13～17のいずれかに記載の楽音情報の通信装置。

【請求項19】 前記出力手段は定期的に第1及び第2の楽音情報の同期をとる請求項13～18のいずれかに記載の楽音情報の通信装置。

【請求項20】 前記出力手段は、第1及び第2の楽音情報の間の時間的ずれの有無を定期的に検出し、時間的ずれを検出したときには該時間的ずれを修正する請求項19記載の楽音情報の通信装置。

【請求項21】 前記第1又は第2の楽音情報は音声データであり、前記出力手段は、前記第1又は第2の楽音情報の間引き又は補間を行うことにより前記時間的ずれを修正する請求項20記載の楽音情報の通信装置。

【請求項22】 前記第1又は第2の楽音情報は複数のMIDIイベントと該複数のMIDIイベント間の時間間隔情報を含み、前記出力手段は、前記時間間隔情報を調整することにより前記時間的ずれを修正する請求項20記載の楽音情報の通信装置。

【請求項23】 (a) 第1及び第2の楽音情報の各々に、共通の時間軸上の時間情報を付与する工程と、(b) 前記第1及び第2の楽音情報の各々と、各楽音情報に付与された時間情報とを関連付けて送信する工程とを含む楽音情報の通信方法。

【請求項24】 前記第1及び第2の楽音情報は異なる楽音情報生成装置により生成された楽音情報である請求項23記載の楽音情報の通信方法。

【請求項25】 前記第1の楽音情報はMIDIデータであり、前記第2の楽音情報は音声データである請求項

2 4 記載の楽音情報の通信方法。

【請求項 2 6】 前記工程 (a) は、第 1 及び第 2 の楽音情報の各々を楽音パケットに分割して各楽音パケットに時間情報を付与して送信パケットを生成し、前記工程 (b) は、前記送信パケットを送信する請求項 2 3 ~ 2 5 のいずれかに記載の楽音情報の通信方法。

【請求項 2 7】 前記工程 (b) はインターネット上に前記送信パケットを送信する請求項 2 6 記載の楽音情報の通信方法。

【請求項 2 8】 さらに、(c) 前記工程 (a) の前に演奏者の演奏操作に応じてリアルタイムで生成された M I D I データ、及び音声データがリアルタイムで電気信号に変換された音声データを入力する工程を含み、前記工程 (b) は、前記入力された M I D I データ及び音声データをリアルタイムで送信する請求項 2 5 記載の楽音情報の通信方法。

【請求項 2 9】 さらに、(d) 前記工程 (b) の後に前記工程 (b) により送信されたものであって前記第 1 及び第 2 の楽音情報及び各楽音情報に関連付けられた時間情報を受信する工程と、

(e) 前記時間情報を基に前記第 1 及び第 2 の楽音情報の同期をとって再生装置に出力する工程とを含む請求項 2 3 ~ 2 8 のいずれかに記載の楽音情報の通信方法。

【請求項 3 0】 前記工程 (e) は、前記受信した第 1 及び第 2 の楽音情報をリアルタイムで再生装置に出力する請求項 2 9 記載の楽音情報の通信方法。

【請求項 3 1】 前記工程 (e) は定期的に第 1 及び第 2 の楽音情報の同期をとる請求項 2 9 又は 3 0 記載の楽音情報の通信方法。

【請求項 3 2】 前記工程 (e) は、第 1 及び第 2 の楽音情報の間の時間的ずれの有無を定期的に検出し、時間的ずれを検出したときには該時間的ずれを修正する請求項 3 1 記載の楽音情報の通信方法。

【請求項 3 3】 前記第 1 又は第 2 の楽音情報は音声データであり、前記工程 (e) は、前記第 1 又は第 2 の楽音情報の間引き又は補間を行うことにより前記時間的ずれを修正する請求項 3 2 記載の楽音情報の通信方法。

【請求項 3 4】 前記第 1 又は第 2 の楽音情報は複数の M I D I イベントと該複数の M I D I イベント間の時間間隔情報を含み、前記工程 (e) は、前記時間間隔情報を調整することにより前記時間的ずれを修正する請求項 3 2 記載の楽音情報の通信方法。

【請求項 3 5】 (a) 第 1 及び第 2 の楽音情報及び各楽音情報に関連付けられた時間情報を受信する工程と、

(b) 前記時間情報を基に前記第 1 及び第 2 の楽音情報の同期をとって再生装置に出力する工程とを含む楽音情報の通信方法。

【請求項 3 6】 前記第 1 及び第 2 の楽音情報は異なる楽音情報生成装置により生成された楽音情報である請求項 3 5 記載の楽音情報の通信方法。

【請求項 3 7】 前記第 1 の楽音情報は M I D I データであり、前記第 2 の楽音情報は音声データである請求項 3 6 記載の楽音情報の通信方法。

【請求項 3 8】 前記工程 (a) は、第 1 及び第 2 の楽音情報の各々が楽音パケットに分割され、各楽音パケットに時間情報が付与された送信パケットを受信する請求項 3 5 ~ 3 7 のいずれかに記載の楽音情報の通信方法。

【請求項 3 9】 前記工程 (a) はインターネット上から前記送信パケットを受信する請求項 3 8 記載の楽音情報の通信方法。

【請求項 4 0】 前記工程 (b) は、前記受信した第 1 及び第 2 の楽音情報をリアルタイムで再生装置に出力する請求項 3 5 ~ 3 9 のいずれかに記載の楽音情報の通信方法。

【請求項 4 1】 前記工程 (b) は定期的に第 1 及び第 2 の楽音情報の同期をとる請求項 3 5 ~ 4 0 のいずれかに記載の楽音情報の通信方法。

【請求項 4 2】 前記工程 (b) は、第 1 及び第 2 の楽音情報の間の時間的ずれの有無を定期的に検出し、時間的ずれを検出したときには該時間的ずれを修正する請求項 4 1 記載の楽音情報の通信方法。

【請求項 4 3】 前記第 1 又は第 2 の楽音情報は音声データであり、前記工程 (b) は、前記第 1 又は第 2 の楽音情報の間引き又は補間を行うことにより前記時間的ずれを修正する請求項 4 2 記載の楽音情報の通信方法。

【請求項 4 4】 前記第 1 又は第 2 の楽音情報は複数の M I D I イベントと該複数の M I D I イベント間の時間間隔情報を含み、前記工程 (b) は、前記時間間隔情報を調整することにより前記時間的ずれを修正する請求項 4 2 記載の楽音情報の通信方法。

【請求項 4 5】 (a) 第 1 及び第 2 の楽音情報の各々に、共通の時間軸上の時間情報を付与する手順と、

(b) 前記第 1 及び第 2 の楽音情報の各々と、各楽音情報に付与された時間情報とを関連付けて送信する手順とをコンピュータに実行させるためのプログラムを記録した媒体。

【請求項 4 6】 前記第 1 及び第 2 の楽音情報は異なる楽音情報生成装置により生成された楽音情報である請求項 4 5 記載のプログラムを記録した媒体。

【請求項 4 7】 前記第 1 の楽音情報は M I D I データであり、前記第 2 の楽音情報は音声データである請求項 4 6 記載のプログラムを記録した媒体。

【請求項 4 8】 前記手順 (a) は、第 1 及び第 2 の楽音情報の各々を楽音パケットに分割して各楽音パケットに時間情報を付与して送信パケットを生成し、前記手順 (b) は、前記送信パケットを送信する請求項 4 5 ~ 4 7 のいずれかに記載のプログラムを記録した媒体。

【請求項 4 9】 前記手順 (b) はインターネット上に前記送信パケットを送信する請求項 4 8 記載のプログラ

ムを記録した媒体。

【請求項50】 さらに、(c)前記手順(a)の前に演奏者の演奏操作に応じてリアルタイムで生成されたMIDIデータ、及び音声データがリアルタイムで電気信号に変換された音声データを入力する手順を含み、

前記手順(b)は、前記入力されたMIDIデータ及び音声データをリアルタイムで送信する請求項47記載のプログラムを記録した媒体。

【請求項51】 さらに、(d)前記手順(b)の後に前記手順(b)により送信されたものであって前記第1及び第2の楽音情報及び各楽音情報に関連付けられた時間情報を受信する手順と、

(e)前記時間情報を基に前記第1及び第2の楽音情報の同期をとって再生装置に出力する手順とを含む請求項45～50のいずれかに記載のプログラムを記録した媒体。

【請求項52】 前記手順(e)は、前記受信した第1及び第2の楽音情報をリアルタイムで再生装置に出力する請求項51記載のプログラムを記録した媒体。

【請求項53】 前記手順(e)は定期的に第1及び第2の楽音情報の同期をとる請求項51又は52記載のプログラムを記録した媒体。

【請求項54】 前記手順(e)は、第1及び第2の楽音情報の間の時間的ずれの有無を定期的に検出し、時間的ずれを検出したときには該時間的ずれを修正する請求項53記載のプログラムを記録した媒体。

【請求項55】 前記第1又は第2の楽音情報は音声データであり、前記手順(e)は、前記第1又は第2の楽音情報の間引き又は補間を行うことにより前記時間的ずれを修正する請求項54記載のプログラムを記録した媒体。

【請求項56】 前記第1又は第2の楽音情報は複数のMIDIイベントと該複数のMIDIイベント間の時間間隔情報を含み、前記手順(e)は、前記時間間隔情報を調整することにより前記時間的ずれを修正する請求項54記載のプログラムを記録した媒体。

【請求項57】 (a)第1及び第2の楽音情報及び各楽音情報に関連付けられた時間情報を受信する手順と、

(b)前記時間情報を基に前記第1及び第2の楽音情報の同期をとって再生装置に出力する手順とをコンピュータに実行させるためのプログラムを記録した媒体。

【請求項58】 前記第1及び第2の楽音情報は異なる楽音情報生成装置により生成された楽音情報である請求項57記載のプログラムを記録した媒体。

【請求項59】 前記第1の楽音情報はMIDIデータであり、前記第2の楽音情報は音声データである請求項58記載のプログラムを記録した媒体。

【請求項60】 前記手順(a)は、第1及び第2の楽音情報の各々が楽音パケットに分割され、各楽音パケットに時間情報が付与された送信パケットを受信する請求

項57～59のいずれかに記載のプログラムを記録した媒体。

【請求項61】 前記手順(a)はインターネット上から前記送信パケットを受信する請求項60記載のプログラムを記録した媒体。

【請求項62】 前記手順(b)は、前記受信した第1及び第2の楽音情報をリアルタイムで再生装置に出力する請求項57～61のいずれかに記載のプログラムを記録した媒体。

10 【請求項63】 前記手順(b)は定期的に第1及び第2の楽音情報の同期をとる請求項57～62のいずれかに記載のプログラムを記録した媒体。

【請求項64】 前記手順(b)は、第1及び第2の楽音情報の間の時間的ずれの有無を定期的に検出し、時間的ずれを検出したときには該時間的ずれを修正する請求項63記載のプログラムを記録した媒体。

【請求項65】 前記第1又は第2の楽音情報は音声データであり、前記手順(b)は、前記第1又は第2の楽音情報の間引き又は補間を行うことにより前記時間的ずれを修正する請求項64記載のプログラムを記録した媒体。

20 【請求項66】 前記第1又は第2の楽音情報は複数のMIDIイベントと該複数のMIDIイベント間の時間間隔情報を含み、前記手順(b)は、前記時間間隔情報を調整することにより前記時間的ずれを修正する請求項64記載のプログラムを記録した媒体。

【請求項67】 楽音パラメータを指定する指定手段と、
前記指定手段により指定される楽音パラメータに応じてMIDIデータ及び音声データを制御する制御手段と、
前記制御手段により制御されるMIDIデータ及び音声データの再生を指示する再生指示手段とを有する楽音情報制御装置。

30 【請求項68】 前記制御手段は、前記指定手段により指定される1つの楽音パラメータに応じてMIDIデータ及び音声データの両方を制御する請求項67記載の楽音情報制御装置。

【請求項69】 前記指定手段は、MIDIデータの楽音パラメータと音声データの楽音パラメータを個別に指定する手段であり、

40 前記制御手段は、前記指定手段により指定されるMIDIデータの楽音パラメータ及び音声データの楽音パラメータに応じてMIDIデータ及び音声データを個別に制御する請求項67記載の楽音情報制御装置。

【請求項70】 さらに、前記指定手段により指定された楽音パラメータを表示する表示手段を有する請求項67～69のいずれかに記載の楽音情報制御装置。

【請求項71】 (a)楽音パラメータを指定する工程と、

50 (b)前記指定された楽音パラメータに応じてMIDI

データ及び音声データを制御する工程と、

(c) 前記制御されたMIDIデータ及び音声データの再生を指示する工程とを有する楽音情報制御方法。

【請求項72】 前記工程(b)は、前記工程(a)で指定された1つの楽音パラメータに応じてMIDIデータ及び音声データの両方を制御する請求項71記載の楽音情報制御方法。

【請求項73】 前記工程(a)は、MIDIデータの楽音パラメータと音声データの楽音パラメータを個別に指定する工程であり、

前記工程(b)は、前記指定されたMIDIデータの楽音パラメータ及び音声データの楽音パラメータに応じてMIDIデータ及び音声データを個別に制御する請求項71記載の楽音情報制御方法。

【請求項74】 さらに、(d)前記工程(a)で指定された楽音パラメータを表示する工程を有する請求項71～73のいずれかに記載の楽音情報制御方法。

【請求項75】 (a) 楽音パラメータを指定する手順と、

(b) 前記指定された楽音パラメータに応じてMIDIデータ及び音声データを制御する手順と、

(c) 前記制御されたMIDIデータ及び音声データの再生を指示する手順とをコンピュータに実行させるためのプログラムを記録した媒体。

【請求項76】 前記手順(b)は、前記手順(a)で指定された1つの楽音パラメータに応じてMIDIデータ及び音声データの両方を制御する請求項75記載のプログラムを記録した媒体。

【請求項77】 前記手順(a)は、MIDIデータの楽音パラメータと音声データの楽音パラメータを個別に指定する手順であり、

前記手順(b)は、前記指定されたMIDIデータの楽音パラメータ及び音声データの楽音パラメータに応じてMIDIデータ及び音声データを個別に制御する請求項75記載のプログラムを記録した媒体。

【請求項78】 さらに、(d)前記手順(a)で指定された楽音パラメータを表示する手順を有する請求項75～77のいずれかに記載のプログラムを記録した媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、楽音情報の処理技術に関し、特に2種類以上の楽音情報を再生するための楽音情報の処理技術に関する。

【0002】

【従来の技術】 電子楽器間の通信の統一規格として、MIDI (musical instrument digital interface) 規格がある。MIDI規格のインターフェースを備えた電子楽器は、MIDI用ケーブルを用いて、他の電子楽器と接続することができる。電子楽器は、MIDI用ケー

ルを介して、MIDIデータを通信することができる。例えば、一の電子楽器は、演奏者が演奏した情報をMIDIデータとして送信し、他の電子楽器は、当該MIDIデータを受信し、楽音を発音することができる。一の電子楽器で演奏すると、他の電子楽器でリアルタイムに発音することができる。

【0003】 また、複数の汎用コンピュータを接続する通信ネットワークでは、種々の情報を通信することができる。例えば、コンピュータに接続されているハードディスク等に音声データ(生の楽音情報)やMIDIデータ等の情報を一度蓄積しておき、通信ネットワークを介して、当該情報を送信する。他のコンピュータは、当該情報を受信して、ハードディスク等の記憶装置に記憶する。汎用の通信ネットワークは、情報の通信を行うのみであり、MIDIとは性質を異にする。

【0004】 MIDI規格は、電子楽器間のリアルタイム通信を可能にするが、長距離の通信及び多数ノード間の通信に適していない。一方、汎用通信ネットワークは、長距離の通信及び多数ノード間の通信に適しているが、電子楽器間のリアルタイム通信を考慮したものではない。

【0005】

【発明が解決しようとする課題】 楽音情報は、MIDIデータと音声データを含む。MIDIデータは、MIDI規格のデータであり、キーオンイベント(例えば押鍵情報)やキーオフイベント(例えば離鍵情報)を含む。音声データは、例えばマイクロフォンを用いて生成されるデータである。マイクロフォンは、音声(楽音を含む)を電気信号に変換してアナログ形式の音声信号を生成することができる。そのアナログ形式の音声信号をデジタル形式に変換することにより、音声データを生成することができる。音声データは、例えば、コンパクトディスクやデジタルオーディオテープに記録されているものである。

【0006】 通信装置は、送信装置と受信装置を含む。送信装置は、MIDIデータと音声データの両方を混在して送信するものが望まれている。受信装置は、MIDIデータと音声データを受信して同時に再生するものが望まれている。MIDIデータ又は音声データのうちのいずれか一方のみを通信することは可能であると考えられる。しかし、MIDIデータと音声データの両方を混在して通信することは困難である。

【0007】 仮に、MIDIデータと音声データの両方を混在して通信できたとしても、両方のデータの同期をとりながら再生を行うことは困難である。両データの再生を同時に開始することは可能であるが、それ以降は同期をとることができない。

【0008】 本発明の目的は、複数の楽音情報の同期をとりながら再生するための楽音情報の通信装置、通信方法又はプログラムを記録した媒体を提供することであ

る。

【0009】本発明の他の目的は、MIDIデータと音声データを適切に制御することができる楽音情報の制御装置、制御方法又はプログラムを記録した媒体を提供することである。

【0010】

【課題を解決するための手段】本発明の一観点によれば、第1及び第2の楽音情報の各々に、共通の時間軸上の時間情報を付与する付与手段と、前記第1及び第2の楽音情報の各々と、各楽音情報に付与された時間情報とを関連付けて送信する送信手段とを有する楽音情報の通信装置が提供される。

【0011】本発明の他の観点によれば、第1及び第2の楽音情報及び各楽音情報に関連付けられた時間情報を受信する受信手段と、前記時間情報を基に前記第1及び第2の楽音情報の同期をとって再生装置に出力する出力手段とを有する楽音情報の通信装置が提供される。

【0012】送信手段は、第1及び第2の楽音情報の各々と、各楽音情報に付与された時間情報とを関連付けて送信し、受信手段はそれらを受信する。出力手段は、上記の時間情報を基に第1及び第2の楽音情報の同期をとって再生装置に出力するので、第1及び第2の楽音情報は互いにタイミングがあって再生される。

【0013】

【発明の実施の形態】図1は、楽音情報の通信ネットワークを示す図である。

【0014】演奏会場1には、MIDI楽器2、音声入力装置12、カメラ4、エンコーダー3、5、及びルータ6が備えられる。音声入力装置12は、例えばマイクロフォンである。演奏会場1では、演奏者がMIDI楽器2を演奏し、歌手がその演奏にあわせて音声入力装置12に向かって歌う。また、音声入力装置12を生ドラムのそばに置いてよい。

【0015】MIDI楽器2は、演奏者の演奏操作に応じてMIDIデータを生成し、エンコーダー3に供給する。音声入力装置12は、歌手の歌声又はドラムの音を電気信号に変換してアナログ形式の音声信号を生成し、リアルタイムでエンコーダー3に供給する。エンコーダー3は、アナログ形式の音声信号をデジタル形式の音声データに変換し、MIDIデータ及び音声データを所定のデータ形式で、ルータ6を介してインターネット上にパケット送信する。データ形式は、後に図5(A)、(B)を参照しながら説明する。

【0016】カメラ4は、演奏者が演奏している様子を撮影し、その様子を画像データとしてエンコーダー5に供給する。エンコーダー5は、画像データを所定のデータ形式で、ルータ6を介してインターネット上にパケット送信する。

【0017】ルータ6は、以下に示すインターネットを介して、MIDIデータ、音声データ及び画像データを

送信する。当該データは、電話回線又は専用回線を通り、ルータ6からサーバー7に供給され、さらに複数のWWW(world wide web)サーバー8に供給される。WWWサーバー8は、いわゆるプロバイダである。

【0018】ユーザは、ホームコンピュータ9をWWWサーバー8に接続することにより、インターネットを使用することができる。ホームコンピュータ9は、インターネットを使用し、MIDIデータ、音声データ及び画像データを受信することができる。ホームコンピュータ9は、ディスプレイ装置とMIDI音源を有し、音声出力装置11に接続される。

【0019】画像データは、ディスプレイ装置に表示される。MIDIデータは、MIDI音源で楽音信号に変換され、音声出力装置11で発音される。音声データは、デジタル形式からアナログ形式に変換されて、音声出力装置11で発音される。ホームコンピュータ9は、MIDIデータと音声データとの同期をとりながら両者を発音させる。同期をとる方法は、後に説明する。演奏会場1の演奏音や歌声と同等の音が音声出力装置11からリアルタイムで発音される。

【0020】また、ホームコンピュータ9の外部に、MIDI音源10を接続すれば、ホームコンピュータ9は、MIDI音源10に楽音信号を生成させ、音声出力装置11から発音させることができる。

【0021】なお、ユーザにとっては、画像データよりもMIDIデータ及び音声データの方が重要な情報であるので、画像データよりもMIDIデータ及び音声データを優先して処理を行う。画像データは、画質が悪く、コマ数が少なくてもさほど気にならないが、MIDIデータ及び音声データに基づく楽音信号は高品質が要求される。

【0022】ユーザは、演奏会場1に赴かなくても、自宅にいながらディスプレイ装置で演奏会場1の模様を見ながら、リアルタイムで演奏及び歌声を聴くことができる。また、自宅のホームコンピュータ9をインターネットに接続すれば、誰でも演奏及び歌声を聴くことができる。例えば、演奏会場1でコンサートを行った場合には、不特定多数人が自宅でそのコンサートを楽しむことができる。

【0023】演奏会場からMIDIデータを自宅に送信することにより、演奏者が複数のユーザのそれぞれの自宅で電子楽器を演奏しているかのような状況を作り出すことができる。MIDIデータの通信は、雑音により音質を下げることはない。

【0024】図2は、送信端であるエンコーダー3と受信端であるホームコンピュータ9を示す。以下、両者の関係を説明するため、便宜的にエンコーダー3をサーバ3と呼び、ホームコンピュータ9をクライアント9と呼ぶ。

【0025】サーバ3とクライアント9は、インターネ

ットのデジタル回線により接続される。サーバ3は、MIDI楽器2からMIDIデータを受け、音声入力装置12からアナログ形式の音声信号を受ける。デジタル形式に変換された音声データ及びMIDIデータは、サーバ3からクライアント9に送信される。クライアント9は、MIDI音源10にMIDIデータを出力し、音声出力装置11にアナログ形式に変換された音声信号を出力する。MIDI音源10は、MIDIデータを受け、アナログ形式の楽音信号を生成して音声出力装置11に出力する。音声出力装置11は、アナログ形式の音声信号及び楽音信号を受けて発音する。

【0026】図3は、図2の具体的なハードウェア構成を示す図である。サーバ3とクライアント9は、共に汎用コンピュータ又はパーソナルコンピュータ等を用いることができる。

【0027】サーバ3とクライアント9は、基本的に同じ構成である。両者の構成を説明する。バス21には、CPU22、RAM24、外部記憶装置25、外部に対してMIDIデータを送受信するためのMIDIインタフェース26、サウンドカード27、ROM28、表示装置29、キーボードやスイッチやマウス等の入力手段30、インターネットを行うための通信インタフェース31が接続されている。

【0028】サウンドカード27は、バッファ27aとコーデック回路27bとクロック27cを有する。バッファ27は、外部に対して入力又は出力するデータをバッファリングする。コーデック回路27bは、A/D変換器及びD/A変換器を有し、アナログ形式とデジタル形式の両者間の変換を行うことができる。クロック27cは、上記の変換を行う際のサンプリングクロックを生成する。サンプリングクロックは、システムクロック23により生成することもできる。その場合は、クロック27cをサウンドカード27に設けなくても良い。

【0029】外部記憶装置25は、例えばハードディスクドライブ、フロッピーディスクドライブ、CD-ROMドライブ、光磁気ディスクドライブ等であり、MIDIデータ、音声データ、画像データ又はコンピュータプログラム等を記憶することができる。

【0030】ROM28は、コンピュータプログラム及び各種パラメータを記憶することができる。RAM24は、バッファやレジスタ等のワーキングエリアを有し、外部記憶装置25に記憶されている内容をコピーして記憶することができる。CPU22は、ROM28又はRAM24に記憶されているコンピュータプログラムに従って、各種演算または処理を行う。CPU22は、システムクロック23から時間情報を得て、タイマ割り込み処理を行うことができる。システムクロック23は、MIDIデータと音声データの同期をとるための時間情報も生成する。

【0031】インターネット回線32には、サーバ3の

通信インタフェース31及びクライアント9の通信インタフェース31が接続される。通信インタフェース31は、インターネットにより、MIDIデータ、音声データ及び画像データを送受信するためのインタフェースである。サーバ3とクライアント9は、インターネット回線32により接続される。

【0032】まず、サーバ3について説明する。MIDIインタフェース26には、MIDI楽器2が接続され、サウンドカード27には、音声入力装置12が接続される。MIDI楽器2は、演奏者の演奏操作に応じてMIDIデータを生成し、MIDIインタフェース26に出力する。音声入力装置12は、演奏会場における音声を入力し、アナログ形式の音声信号をサウンドカード27に出力する。サウンドカード27は、アナログ形式の音声信号をデジタル形式の音声データに変換する。

【0033】図4(A)に示すように、サーバ3のRAM24は、MIDIデータ用送信バッファ24aと音声データ用送信バッファ24bを有する。MIDIインタフェース26に入力されたMIDIデータは、MIDIデータ用送信バッファ24aに蓄積され、サウンドカード27に入力された音声データは、音声データ用送信バッファ24bに蓄積される。CPU22は、上記のMIDIデータと音声データをそれぞれ送信バッファ24aと24bから読み出して、通信インタフェース31を介してインターネット回線32上にパケット送信する。

【0034】次に、クライアント9について説明する。図3に示すように、MIDIインタフェース26には、MIDI音源10が接続され、サウンドカード27には、音声出力装置12が接続される。CPU22は、通信インタフェース31を介して、インターネット回線32上からMIDIデータと音声データと画像データを受信する。

【0035】図4(B)に示すように、クライアント9のRAM24は、MIDIデータ用受信バッファ24c、音声データ用受信バッファ24d、MIDIデータ用再生バッファ24e、及び音声データ用再生バッファ24fを有する。MIDIデータ用受信バッファ24cに、受信したMIDIデータをバッファリングし、音声データ用受信バッファ24dに、受信した音声データをバッファリングする。

【0036】CPU22は、MIDIデータ用受信バッファ24c内のMIDIデータをMIDIデータ用再生バッファ24eに記憶し、音声データ用受信バッファ24d内の音声データを音声データ用再生バッファ24fに記憶する。MIDIデータ又は音声データは、時間の経過と共に、再生バッファ24e又は24fのアドレスの小さい方から順に記憶される。再生バッファ24e、24fの各アドレスは、時間情報をも表す。すなわち、アドレス軸が時間軸に対応する。

【0037】再生バッファ24e内のMIDIデータ

は、MIDIインタフェース26からMIDI音源10に出力される。MIDI音源10は、MIDIデータを受け取り、楽音信号を生成して音声出力装置11に出力する。サウンドカード27は、再生バッファ24f内のデジタル形式の音声データをアナログ形式の音声信号に変換して、音声出力装置11に出力する。音声出力装置11は、上記の楽音信号と音声信号を受け取り、発音する。

【0038】通信インタフェース31は、インターネット用インタフェースの他、イーサネット用インタフェース、IEEE1394規格のデジタル通信インタフェース、RS-232C用インタフェースでもよく、種々のネットワークに接続することができる。

【0039】サーバ3は、MIDIデータ等を送信するためのコンピュータプログラムを記憶する。クライアント9は、MIDIデータ等を受信するためのコンピュータプログラムを記憶する。コンピュータプログラムや各種パラメータ等を外部記憶装置25に記憶させておき、それをRAM24に読み込むことにより、コンピュータプログラム等の追加やバージョンアップ等が容易に行える。

【0040】CD-ROM（コンパクトディスク・リード・オンリィ・メモリ）ドライブは、CD-ROMに記憶されているコンピュータプログラムや各種データを読み出す装置である。読み出したコンピュータプログラムや各種データは、ハードディスクにストアされる。コンピュータプログラムの新規インストールやバージョンアップ等が容易に行える。

【0041】通信インターフェース31はLAN（ローカルエリアネットワーク）やインターネット、電話回路等の通信ネットワーク32に接続されており、該通信ネットワーク32を介して、コンピュータ33と接続される。外部記憶装置25内に上記のコンピュータプログラム等が記憶されていない場合、コンピュータ33からコンピュータプログラム等をダウンロードすることができる。サーバ3又はクライアント9は、通信インターフェース31及び通信ネットワーク32を介してコンピュータ33へコンピュータプログラム等のダウンロードを要求するコマンドを送信する。コンピュータ33は、このコマンドを受け、要求されたコンピュータプログラム等を、通信ネットワーク32を介してサーバ3又はクライアント9へ配信する。サーバ3又はクライアント9が通信インタフェース31を介して、コンピュータプログラム等を受信して外部記憶装置25に蓄積することにより、ダウンロードが完了する。

【0042】なお、本実施例は、本実施例に対応するコンピュータプログラム等をインストールした市販のパーソナルコンピュータ等によって、実施させるようにしてもよい。その場合には、本実施例に対応するコンピュータプログラム等を、CD-ROMやフロッピディスク等

の、コンピュータが読み込むことができる記憶媒体に記憶させた状態で、ユーザーに提供してもよい。そのパーソナルコンピュータ等が、LAN、インターネット、電話回線等の通信ネットワークに接続されている場合には、通信ネットワークを介して、コンピュータプログラムや各種データ等をパーソナルコンピュータ等に提供してもよい。

【0043】また、サーバ3又はクライアント9は、パーソナルコンピュータの他、電子楽器、ゲーム機、カラオケ装置、テレビ等の形態として適用してもよい。

【0044】図5（A）は、サーバ3が送信するMIDIデータパケット49の構造を示す。

【0045】MIDIデータパケット49は、同期をとるためのタイムスタンプ41、当該パケットがMIDIデータであることを示す識別子（ID）42、当該パケットのサイズ43、MIDIデータ44を有する。

【0046】タイムスタンプ41は、パケット内のMIDIデータ44の送信時刻を表すと共に、演奏時刻、録音時刻、再生時刻をも表す。識別子42は、MIDIデータパケットや音声データパケットや画像データパケット等のパケットの種類を表す。

【0047】MIDIデータ44は、スタンダードMIDIファイルフォーマットに準拠したものであり、デルタタイム（インターバル）とMIDIイベントを1組にしたデータの列である。デルタタイム46がMIDIイベント45と47の間にある場合、デルタタイム46は、前のMIDIイベント45と次のMIDIイベント47の間の時間間隔を表す。ただし、デルタタイムが0であるときには、デルタタイムを省略することができる。

【0048】図5（B）は、サーバ3が送信する音声データパケット50の構造を示す。音声データパケット50は、同期をとるためのタイムスタンプ41、当該パケットが音声データであることを示す識別子（ID）42、当該パケットのサイズ43、デジタル音声データ48を有する。

【0049】タイムスタンプ41は、MIDIデータパケットの場合と同様に、パケット内の音声データ48の録音時刻等を表す。デジタル音声データ48は、音声入力装置12（図3）を基に生成されたデータである。

【0050】次に、MIDIデータと音声データの同期のとり方を説明する。MIDIデータパケット49は、タイムスタンプ41を有する。音声データパケット50も、タイムスタンプ41を有する。

【0051】両者のタイムスタンプ41は、共にシステムクロック23（図3）により生成される共通の時間軸上の時間情報であり、演奏開始からの経過時間である。例えば、コンサートを21時00分に開始し22時00分に終了したときには、タイムスタンプはコンサート開始時が0時0分0秒00（21時00分）であり、コン

サート終了時が1時0分0秒00（22時00分）である。

【0052】クライアント9は、サーバ3から受信したタイムスタンプ41を基に時間情報のカウントを行えば、演奏開始時からの経過時間を知ることができる。この経過時間を利用することにより、MIDIデータと音声データの同期をとることができる。

【0053】MIDIデータと音声データの再生開始時に同期をとっても、その後にタイミングがずれて行くことがある。例えば、MIDIデータの生成クロックと音声データの生成クロックが非同期又は異なることがある。具体的には、図3において、MIDIインタフェース26がMIDI楽器2からMIDIデータを受信するクロック23と、サウンドカード27がA/D変換するクロック27cが非同期又は異なることがある。その場合は、MIDIデータと音声データの再生開始時に同期をとっても、その後、時間の経過と共に、両者のタイミングがずれていく。それと同時に、サーバ3でのタイミングとクライアント9でのタイミングがずれていく。

【0054】同様に、サーバ3のクロックとクライアント9のクロックとに誤差がある場合にも、上記のタイミングのずれが生じることがある。

【0055】次に、上記のタイミングのずれを軽減又はなくすために、再生開始時だけでなく、その後も定期的に同期をとる方法を説明する。

【0056】まず、MIDIデータを定期的に同期させる方法を説明する。MIDIデータパケット内のタイムスタンプを初期値として所定時間間隔でカウントすることにより、現在のMIDIデータの再生経過時間を獲得することができる。MIDIデータ用再生バッファ24e（図4（B））のアドレスも再生経過時間に対応しているが、この時間が上記の時間とずれることがある。そこで、そのずれを修正することにより、MIDIデータの同期をとる。MIDIデータのタイミングの修正方法は、後に図7を参照しながら説明する。

【0057】同様に、音声データパケット内のタイムスタンプを初期値としてカウントすることにより、現在の音声データの再生経過時間を獲得することができる。音声データ用再生バッファ24f（図4（B））のアドレスも再生経過時間に対応しているが、この時間が上記の時間とずれることがある。そこで、そのずれを修正することにより、音声データの同期をとる。音声データのタイミングの修正方法は、図6を参照しながら説明する。

【0058】図6は、音声データのタイミングの修正方法を示す。横軸は時間である。音声データ及びMIDIデータの同期は例えば2秒間隔でとられる。すなわち、2秒毎に再生経過時間にずれが生じているか否かをチェックし、ずれている場合にはそのずれを修正する。

【0059】音声データDD1は、本来の再生経過時間に従った2秒間のデータであり、時刻t1に開始して時

刻t3に終了する。サーバ3は、例えば0.2秒間分の各音声データD1～D10をそれぞれ1パケットとしてパケット通信する。音声データのサンプリングレートが例えば50kHzであるとする、0.02ms毎にサンプリングが行われる。各音声データD1～D10は、0.2秒間のデータであるので、10,000ポイント（サンプル）を有する。

【0060】例えば、実際に再生すべき音声データDD2が、本来の音声データDD1に比べ、0.1秒間だけ遅れて時刻t2から開始するとする。この場合、音声データDD2を音声データDD3に修正して再生する。すなわち、音声データDD2内の各データD1～D10から50ポイントずつ間引きし、音声データDD3を生成する。具体的には、20ポイント毎に1ポイントの間引きげばよい。音声データDD3内の各データD1～D10は、9,500ポイントを有し、再生時間が0.19秒に減少する。10個のデータD1～D10を有する音声データDD3は、再生時間が1.9秒である。

【0061】音声データDD3は、音声データDD2に比べ、再生時間が0.1秒だけ短いので、音声データの遅れを回復することができる。すなわち、音声データDD3は、音声データDD1に比べ、開始時（時刻t2）には0.1秒遅れていたが、終了時（時刻t3）には遅れがなくなる。2秒間隔で同期をとる場合には、2秒の間に遅れを取り戻せばよい。

【0062】逆に、0.1秒進んだ場合には、20ポイント毎に1ポイントを補間して、1パケットのデータを10,500ポイントにすればよい。補間は、例えば、前後のデータを平均する。

【0063】図7（A）～（C）は、MIDIデータのタイミングの修正方法を示す。MIDIデータは、図5（A）のMIDIデータ44に対応する。MIDIデータと音声データの同期は例えば2秒間隔でとられる。

【0064】図7（A）に示すように、MIDIデータDD1は、MIDIイベントEV1、デルタタイム（0.5秒）DT1、MIDIイベントEV2、デルタタイム（1.5秒）DT2を順次有する。MIDIデータDD1の再生時間は、デルタタイム（0.5秒）DT1とデルタタイム（1.5秒）DT2を加算した値であり、2秒である。

【0065】MIDIデータDD1が0.1秒遅れた場合、図7（B）に示すMIDIデータDD2に修正する。MIDIデータDD2は、デルタタイムDT1が0.5-0.1×1/4秒であり、デルタタイムDT2が1.5-0.1×3/4秒である。すなわち、デルタタイムDT1とDT2の時間配分に応じて、各デルタタイムDT1、DT2を修正する。デルタタイムDT1は、 $0.1 \times 1 / 4 (= 0.1 \times 0.5 / (0.5 + 1.5))$ 秒だけ早め、デルタタイムDT2は、 $0.1 \times 3 / 4 (= 0.1 \times 1.5 / (0.5 + 1.5))$ 秒

だけ早める。MIDIデータDD2は、MIDIデータDD1に比べ、0.1秒だけ短くなり、1.9秒となる。MIDIデータDD2は、0.1秒の遅れを取り戻すことができる。

【0066】逆に、MIDIデータDD1が0.1秒進んだ場合、図7(C)に示すMIDIデータDD3に修正する。MIDIデータDD3は、デルタタイムDT1が $0.5 + 0.1 \times 1 / 4$ 秒であり、デルタタイムDT2が $1.5 + 0.1 \times 3 / 4$ 秒である。MIDIデータDD3は、MIDIデータDD1に比べ、0.1秒だけ長くなり、2.1秒となる。MIDIデータDD3は、0.1秒の進みを修正することができる。

【0067】図8は、サーバ3の第1の処理を示すフローチャートである。ステップSA1では、MIDIイベントを取得する。具体的には、MIDI楽器からMIDIインタフェースを介してMIDIイベントを取得する。

【0068】ステップSA2では、当該MIDIイベントがパケットデータの先頭イベントであるか否かをチェックする。すなわち、パケット送信する際に、パケット内の先頭イベントに該当するか否かをチェックする。先頭イベントであるときには、yesの矢印に従い、ステップSA3へ進む。先頭イベントでないときには、noの矢印に従い、ステップSA3をバイパスしてステップSA4へ進む。

【0069】ステップSA3では、タイムスタンプを付与する。タイムスタンプは、演奏開始時（録音開始時）からの経過時間であり、パケット内のデータの演奏時刻に相当する。その後、ステップSA4へ進む。

【0070】ステップSA4では、デルタタイムを付与する。デルタタイムは、前のMIDIイベントと次のMIDIイベントとの間の時間間隔である。

【0071】ステップSA5では、取得したMIDIイベントをデルタタイム又はタイムスタンプと共に送信バッファ24a（図4(A)）に順次格納する。当該MIDIイベントがパケット内の先頭イベントであるときには、タイムスタンプ、デルタタイム、取得したMIDIイベントを送信バッファに格納する。当該MIDIイベントがパケット内の先頭イベントでないときには、デルタタイム、取得したMIDIイベントを送信バッファに格納する。

【0072】ステップSA6では、所定のMIDIイベント数を越えたか、あるいは所定時間経過したか否かの条件をチェックする。上記のイベント数又は時間により、パケットの大きさが決まる。条件を満たさない場合には、noの矢印に従い、第1の処理を終了し、その後、次のMIDIイベントが入力されれば、ステップSA1から始まる上記の処理を繰り返す。ステップSA6の条件を満たす場合には、yesの矢印に従い、ステップSA7へ進む。

【0073】ステップSA7では、パケットを送信バッファから読み出し配信する。具体的には、図5(A)に示すように、タイムスタンプ41、識別子42、パケットサイズ43、MIDIデータ44を配信する。パケットのサイズは、例えば約500バイトである。その後、サーバ3は第1の処理を終了する。

【0074】図9は、サーバ3の第2の処理を示すフローチャートである。ステップSB1では、音声入力装置から音声信号を取得する。

10 【0075】ステップSB2では、所定のサンプリングレートにて音声信号をサンプリングする。具体的には、サウンドカードが音声信号をA/D変換して、デジタル音声データを生成する。

【0076】ステップSB3では、当該音声データがパケットデータの先頭データであるか否かをチェックする。先頭データであるときには、yesの矢印に従い、ステップSB4へ進む。先頭データでないときには、noの矢印に従い、ステップSB4をバイパスしてステップSB5へ進む。

20 【0077】ステップSB4では、タイムスタンプを付与する。タイムスタンプは、演奏開始時（録音開始時）からの経過時間であり、パケット内のデータの演奏時刻に相当する。その後、ステップSB5へ進む。

【0078】ステップSB5では、サンプリングした音声データを送信バッファ24b（図4(A)）に順次格納する。ただし、当該音声データがパケット内の先頭データであるときには、タイムスタンプと当該音声データを送信バッファに格納する。

30 【0079】ステップSB6では、所定のデータ量を越えたか、あるいは所定時間経過したか否かの条件をチェックする。条件を満たさない場合には、noの矢印に従い、第2の処理を終了し、その後、次の音声信号が入力されれば、ステップSB1から始まる上記の処理を繰り返す。ステップSB6の条件を満たす場合には、yesの矢印に従い、ステップSB7へ進む。

40 【0080】ステップSB7では、パケットを送信バッファから読み出し配信する。具体的には、図5(B)に示すように、タイムスタンプ41、識別子42、パケットサイズ43、音声データ48を配信する。その後、サーバ3は第2の処理を終了する。

【0081】図10は、クライアント9のパケット受信処理を示すフローチャートである。ステップSC1では、通信インタフェースを介してパケットデータを受信する。

【0082】ステップSC2では、クロックカウンタにパケット内のタイムスタンプの値をセットする。その後、クライアント9のシステムクロックがクロックカウンタの値を定期的にインクリメントする。

50 【0083】ステップSC3では、タイムスタンプが0であるか否かをチェックする。最初のパケットを受信し

たときには、そのパケット内のタイムスタンプは0である。0であるときには、yesの矢印に従い、ステップSC4へ進む。0でないときには、noの矢印に従い、ステップSC4をバイパスしてステップSC5へ進む。

【0084】ステップSC4では、スケジューラを起動し、上記のクロックカウンタのクロック計測を開始する。スケジューラは、MIDIデータと音声データの同期をとるための割り込み処理であり、詳細は、後に図14を参照しながら説明する。上記のクロック計測は、図12に示すように、所定時間間隔の割り込み処理により行われる。ステップSE1では、クロックカウンタの値をインクリメントする。クロックカウンタの値は、図10のステップSC2において設定され、その後、所定時間間隔でインクリメントされる。ステップSE2及びSE3は後に説明する。その後、図10のステップSC5へ進む。

【0085】ステップSC5では、パケット内の識別子(ID)がMIDIデータパケットを示すものであるか否かをチェックする。MIDIデータパケットを示すものであるときには、MIDIデータパケットの処理を行うため、yesの矢印に従い、ステップSC6へ進む。一方、MIDIデータパケットを示すものでないときには、音声データパケットを示すものであることを意味し、noの矢印に従い、ステップSC9へ進む。

【0086】ステップSC6では、受信したパケットデータをMIDIデータ用受信バッファ24c(図4(B))に格納する。

【0087】次に、ステップSC7では、MIDIデータ用再生バッファ24e(図4(B))に上記の受信バッファ内のパケットを転送する。MIDIデータ用再生バッファは、MIDI再生モジュールが再生処理するためのバッファであり、アドレスが時間に相当する。

【0088】その後、ステップSC8でMIDI再生モジュール処理を行う。MIDI再生モジュール処理の詳細は、後に図11のフローチャートを参照しながら説明する。

【0089】次に、音声データの処理を説明する。ステップSC9は、受信したパケットデータを音声データ用受信バッファ24d(図4(B))に格納する。その後、ステップSC10へ進む。

【0090】ステップSC10では、音声データ用再生バッファ24f(図4(B))に上記の受信バッファ内のパケットを転送する。音声データ用再生バッファは、音声再生モジュールが再生処理するためのバッファであり、アドレスが時間に相当する。

【0091】その後、ステップSC11で音声再生モジュール処理を行う。音声再生モジュール処理の詳細は、後に図13のフローチャートを参照しながら説明する。

【0092】図11は、図10のステップSC8におけるMIDI再生モジュール処理の詳細を示すフローチャ

ートである。当該処理は、シーケンサと同様に、MIDIイベントの再生処理を行う。

【0093】ステップSD1では、デルタタイムのカウント値が0か否かをチェックする。まず、パケット内のデルタタイムが読み出され、カウンタにセットされる。その後、図12の割り込み処理により、そのデルタタイムは、クライアント9のシステムクロックにより減算される。具体的には、ステップSE2で、デルタタイムのカウント値が0か否かをチェックする。0でないときには、noの矢印に従い、ステップSE3において、デルタタイムのカウント値をデクリメントし、割り込み前の処理に戻る。一方、0であるときには、yesの矢印に従い、ステップSE3をバイパスして割り込み前の処理に戻る。

【0094】図11に戻り、ステップSD1でデルタタイムのカウント値が0でないときには、未だ再生タイミングに到達していないことを意味するので、MIDI再生モジュール処理を終了する。

【0095】一方、ステップSD1でデルタタイムのカウント値が0であると判断されたときには、再生タイミングに到達したことを意味するので、ステップSD2へ進む。

【0096】ステップSD2では、上記の再生バッファより読み出したMIDIイベントをMIDI音源に転送する。MIDI音源は、MIDIイベントを基に楽音信号を生成し、音声出力装置が当該楽音信号を基に発音する。

【0097】ステップSD3では、次のイベントを再生バッファから読み出す。当該イベントは、MIDIイベント又はデルタタイムである。

【0098】ステップSD4では、読み出したイベントがデルタタイムであるか否かをチェックする。デルタタイムでないときには、MIDIイベントであることを意味し、noの矢印に従い、ステップSD2へ戻り、読み出したMIDIイベントをMIDI音源に転送する。一方、読み出したイベントがデルタタイムであるときには、yesの矢印に従い、ステップSD5へ進む。

【0099】ステップSD5では、読み出したデルタタイムをデルタタイムカウンタにセットする。デルタタイムカウンタの値は、図12の割り込み処理においてデクリメントされる。その後、MIDI再生モジュール処理を終了する。

【0100】なお、上記のMIDI再生モジュール処理は、MIDIデータパケットを受信したときのみに行うのではなく、割り込み処理により定期的にも行われる。MIDI再生モジュール処理を定期的に行い、その都度再生バッファ内に存在するデータを再生処理することにより、所定の分解能で再生処理を行うことができる。

【0101】図13は、図10のステップSC11における音声再生モジュール処理の詳細を示すフローチャ

トである。

【0102】ステップSF1では、音声データ用再生バッファよりパケットデータを所定サンプル分だけサウンドカードへ転送する。サウンドカードは、音声データをアナログ形式の音声信号に変換する。音声出力装置は、当該音声信号を基に発音する。その後、音声再生モジュール処理を終了する。

【0103】なお、上記の音声再生モジュール処理は、音声データパケットを受信したときのみに行うのではなく、割り込み処理により定期的にも行われる。音声再生モジュール処理を定期的に行い、その都度再生バッファ内に存在するデータを再生処理することにより、所定の分解能で再生処理を行うことができる。

【0104】図14は、図10のステップSC4における第1のスケジュー処理の詳細を示すフローチャートである。当該処理は、割り込み処理により定期的（例えば2秒間隔）に起動され、MIDIデータと音声データの同期をとる。

【0105】ステップSG1では、音声データの再生時間を算出する。まず、音声データ用再生バッファのリードポインタ（アドレス）を獲得し、時間情報に換算する。当該再生バッファは、音声再生モジュール（図13）が再生処理するためのバッファであり、アドレスが再生時間に対応している。次に、上記のリードポインタから最近前のタイムスタンプを獲得する。次に、上記のタイムスタンプとリードポインタが示す時間とを加算し、再生時間を算出する。

【0106】ステップSG2では、上記の再生時間とクロックカウンタの値を比較する。クロックカウンタの値は、まず、図10のステップSC2でタイムスタンプの値にセットされ、その後図12の割り込み処理でインクリメントされる。比較の結果、両者が同じであれば、音声データのタイミングが合っており、両者が異なれば、音声データのタイミングがずれていることを意味する。

【0107】ステップSG3では、上記の比較結果に応じ、図6に示すように、音声データのポイント数（サンプル数）を制御し、音声データの時間的ずれを修正する。すなわち、上記のリードポインタから次回スケジューが起動されるまでに再生されるデータ間のポイント数を制御する。

【0108】ステップSG4では、MIDIデータの再生時間を算出する。まず、MIDIデータ用再生バッファのリードポインタ（アドレス）を獲得し、時間情報に換算する。当該再生バッファは、MIDI再生モジュール（図11）が再生処理するためのバッファであり、アドレスが再生時間に対応している。次に、上記のリードポインタから最近前のタイムスタンプを獲得する。次に、上記のタイムスタンプとリードポインタが示す時間とを加算し、再生時間を算出する。

【0109】ステップSG5では、上記の再生時間とク

ロックカウンタの値を比較する。比較の結果、両者が異なれば、MIDIデータのタイミングがずれており、両者が同じであれば、MIDIデータのタイミングが合っていることを意味する。

【0110】ステップSG6では、上記の比較結果に応じ、図7に示すように、MIDIデータ中のデルタタイム値を制御し、MIDIデータの時間的ずれを修正する。すなわち、上記のリードポインタから次回スケジューが起動されるまでに再生されるデータ間のデルタタイム値を制御する。その後、第1のスケジュー処理を終了する。

【0111】上記のように、音声データの時間的ずれとMIDIデータの時間的ずれを定期的に修正することにより、音声データとMIDIデータの同期をとることができる。また、音声データ及びMIDIデータに関して、サーバ3とクライアント9との間でも定期的に同期をとることができる。

【0112】次に、クライアント9において音声データとMIDIデータの同期をとる簡便な方法を図15と図16に2種類示す。これらの方法は、サーバ3とクライアント9との間で定期的に同期をとることはしないが、音声データとMIDIデータの間の同期はとられるので、両者間のタイミングのずれが解消される。

【0113】図15は、図14の第1のスケジュー処理に代わる第2のスケジュー処理を示すフローチャートである。当該処理は、割り込み処理により定期的（例えば2秒間隔）に起動され、MIDIデータと音声データの同期をとる。

【0114】ステップSH1では、上記のステップSG1と同様に、リードポインタとタイムスタンプとから音声データの再生時間を算出する。

【0115】ステップSH2では、上記のステップSG4と同様に、リードポインタとタイムスタンプとからMIDIデータの再生時間を算出する。

【0116】ステップSH3では、上記の音声データの再生時間とMIDIデータの再生時間を比較する。比較の結果、両者が同じであれば、両者のタイミングが合っており、両者が異なれば、両者のタイミングがずれていることを意味する。両者を比較することにより、音声データとMIDIデータとの同期をとることができる。ただし、図14の第1のスケジュー処理と異なり、クロックカウンタ値との比較を行っていないので、サーバ3とクライアント9との間の同期はとっていない。

【0117】ステップSH4では、上記のステップSG6と同様に、比較の結果に応じて、MIDIデータ中のデルタタイム値を制御し、MIDIデータの時間的ずれを修正する。MIDIデータの時間的ずれを定期的に修正することにより、音声データとMIDIデータの間の同期をとることができる。その後、第2のスケジュー処理を終了する。

【0118】図16は、図14の第1のスケジューラ処理に代わる第3のスケジューラ処理を示すフローチャートである。当該処理は、割り込み処理により定期的（例えば2秒間隔）に起動され、MIDIデータと音声データの同期をとる。

【0119】第2のスケジューラ処理（図15）では、MIDIデータのタイミングを修正することにより、MIDIデータと音声データの同期をとっていたが、第3のスケジューラ処理（図16）では、音声データのタイミングを修正することにより、MIDIデータと音声データの同期をとる。

【0120】ステップS11では、上記のステップSG1と同様に、リードポイントとタイムスタンプとから音声データの再生時間を算出する。

【0121】ステップS12では、上記のステップSG4と同様に、リードポイントとタイムスタンプとからMIDIデータの再生時間を算出する。

【0122】ステップS13では、上記の音声データの再生時間とMIDIデータの再生時間を比較する。比較の結果、両者が同じであれば、両者のタイミングが合っており、両者が異なれば、両者のタイミングがずれていることを意味する。

【0123】ステップS14では、上記のステップSG3と同様に、比較の結果に応じて、音声データのポイント数（サンプル数）を制御し、音声データの時間的ずれを修正する。音声データの時間的ずれを定期的に修正することにより、音声データとMIDIデータの同期をとることができる。その後、第3のスケジューラ処理を終了する。

【0124】以上のように、本実施例では、2種類以上の異なる楽音情報（例えば音声データとMIDIデータ）をパケット送信する際に、各パケットにタイムスタンプを付与する。タイムスタンプは、パケット内の楽音情報の発生時間（再生時間）を示す。なお、異種の楽音情報に限定されず、同種の2以上の楽音情報を通信する際にも適用することができる。

【0125】受信装置であるクライアント9は、タイムスタンプを含むパケットを受信する。パケット内の楽音情報を再生する際には、このタイムスタンプを利用することにより、音声データとMIDIデータの同期をとることができる。具体的には、音声データとMIDIデータとの間のタイミングのずれを定期的に検出し、ずれがある場合にはそのずれを修正することにより、両者の同期をとる。

【0126】クライアント9は、クロックカウンタにタイムスタンプ値をセットし、その値をインクリメントすることにより、クライアント9におけるクロックカウンタ値を計測することができる。このクロックカウンタ値を基にMIDIデータ及び音声データの時間的ずれを検出すれば、サーバ3とクライアント9の間の同期をとる

こともできる。

【0127】なお、本実施例は、音声データ及びMIDIデータをインターネットで通信する場合に限定されない。例えば、IEEE1394規格のデジタルシリアル通信や通信衛星等の他の通信にも適用することができる。

【0128】図17は、ホームコンピュータ9の表示装置29（図1及び図3）に表示される入力画面を示す。ユーザは、ホームコンピュータ9のマウスやキーボードを用いて、その入力画面で以下の入力を行うことができる。

【0129】入力画面には、ボリューム操作子61、バランス操作子62、再生ボタン63、停止ボタン64、MIDIデータ表示ランプ65、音声データ（以下、オーディオデータという）表示ランプ66、その他の操作ボタン67が表示される。

【0130】ボリューム操作子61は、MIDIデータ及びオーディオデータの音量を指定するための操作子である。例えば、マウスカーソルをボリューム操作子61の位置に移動させ、マウスのドラッグ操作を行うことにより、MIDIデータ及びオーディオデータの音量を変えることができる。ボリューム操作子61を上に移動させれば音量を上げることができ、ボリューム操作子61を下に移動させれば音量を下げるができる。その操作に応じて、ボリューム操作子61の表示位置も変わる。ユーザは、ボリューム操作子61の位置を参照することにより、音量の大きさを知ることができる。ボリューム操作子61の詳細は、後に図18（A）、（B）を参照しながら説明する。

【0131】バランス操作子62は、MIDIデータとオーディオデータの音量バランスを指定するための操作子である。例えば、バランス操作子62を上に移動させれば、オーディオデータをMIDIデータに比較して大きくすることができ、バランス操作子62を下に移動させれば、MIDIデータをオーディオデータに比較して大きくすることができる。バランス操作子62の詳細は、後に図19を参照しながら説明する。

【0132】再生ボタン63は、MIDIデータ及び／又はオーディオデータの再生を指示するためのボタンであり、停止ボタン64は、その再生の停止を指示するためのボタンである。

【0133】再生ボタン63及び停止ボタン64は、MIDIデータとオーディオデータとを別々にスタート及びストップさせるようにしてよいし、両方を同時にスタート及びストップさせるようにしてもよい。別々にスタート及びストップさせる場合には、再生ボタン63と停止ボタン64を、MIDIデータ用のものとオーディオデータ用のものとに分けて別に設けてもよい。

【0134】MIDIデータ表示ランプ65は、MIDIデータの再生中であることを知らせるためのランプで

あり、オーディオデータ表示ランプ66は、オーディオデータの再生中であることを知らせるためのランプである。

【0135】図18(A)は、ボリューム操作子61による第1の音量制御を説明するための図である。

【0136】ボリューム操作子61を操作することにより、ボリューム係数 α を制御することができる。すなわち、ボリューム操作子61の操作位置に応じて、ボリューム係数 α が決まる。ボリューム操作子61は、最上位置が係数 $\alpha=1$ であり、最下位置が係数 $\alpha=0$ であり、中間位置が係数 $\alpha=0.5$ である。

【0137】その最上位置と最下位置との間の距離を y とし、最下位置から現在の操作子61の位置までの距離を x とすると、現在の操作子61の位置に対応する係数 α は、次式により求められる。

【0138】 $\alpha = x / y$

次に、MIDIデータの音量の制御方法を説明する。ホームコンピュータは、上記のように、演奏会場からインタネットを介してMIDIデータを受信する。そのMIDIデータの中には、トラックボリューム（メインボリューム）が含まれる。トラックボリュームは、MIDIのコントロールチェンジメッセージにより指定することができる。コントロールチェンジメッセージの第1バイトデータを「7」に設定し、第2バイトデータに0から127までのトラックボリューム値Vtrを設定することができる。

【0139】実際のMIDIデータのボリュームVtr1は、上記のトラックボリュームVtrにボリューム係数 α を乗じた値とする。

【0140】 $Vtr1 = Vtr \times \alpha$

トラックボリュームVtrは、最大値127にしてもよいが、それでは演奏者の意思が反映されないので、上記のように、演奏者がトラックボリュームVtrとして送信した値を採用するのが好ましい。

【0141】次に、オーディオデータの音量の制御方法を説明する。実際のオーディオデータのボリュームVau1は、音声出力装置11（図1）の最大ボリュームVauにボリューム係数 α を乗じた値とする。

【0142】 $Vau1 = Vau \times \alpha$

MIDIデータとオーディオデータは、上記の方法により、同期をとることができるが、その際、同時期のMIDIデータとオーディオデータとは音楽上の相関関係がある。例えば、音量、エフェクト情報（イコライザやフィルタ等）等の相関関係がある。そのため、1つのボリューム操作子61で、MIDIデータとオーディオデータの両方の音量を同時に制御することの必要性が生じる。ボリューム操作子61と同様に、エフェクト情報等の操作子を設けてもよい。

【0143】なお、1つのボリューム操作子61を操作することにより、全MIDIチャンネルのMIDIデータ

及びオーディオデータの音量を制御してもよいし、ボリューム操作子61を2つ設け、MIDIデータとオーディオデータとを別々に制御するようにしてもよい。また、MIDIチャンネル数分のボリューム操作子61を設けて、MIDIチャンネル毎に別々に音量を制御してもよい。その場合、コントロールチェンジメッセージには、MIDIチャンネル番号が含まれているので、MIDIチャンネル毎に別のトラックボリュームVtrを設定することができる。

10 【0144】MIDIチャンネルの数だけボリューム操作子61を表示させると、数が多すぎて、却って邪魔になる場合がある。その場合には、全てのMIDIチャンネルのボリューム操作子61を表示するのではなく、あるMIDIチャンネルのMIDIデータを受信したときだけ（又は再生しているときだけ）、そのMIDIチャンネルのボリューム操作子を表示するようにしてもよい。これにより、余分なボリューム操作子61の表示をなくことができ、さらに、以下の効果も生じる。

20 【0145】すなわち、あるMIDIチャンネルのボリューム操作子61を操作しても、そのMIDIチャンネルのMIDIデータが存在しないために、操作子61による効果が現れないというユーザの混乱を防止することができる。

【0146】次に、ボリューム係数 α の別の求め方を示す。図18(B)は、ボリューム操作子61による第2の音量制御を説明するための図である。

【0147】ボリューム操作子61の全操作領域を、例えば4分割し、5つのポイントを設定する。そして、5つのポイントにそれぞれ係数 $\alpha=0, 0.25, 0.5, 0.75, 1$ を割り当てる。5つのポイントの中で、ボリューム操作子61が一番近いポイントを検出し、そのポイントに対応する係数 α が選択されて設定される。例えば、ボリューム操作子61が一番近いポイントの係数 $\alpha=0.75$ が設定される。

【0148】ボリューム操作子61が一番近いポイントを検出する方法の例を示す。まず、ボリューム操作子61から5つのポイントまでのそれぞれの距離を算出する。そして、その5つの距離の中で一番短い距離のポイントを見つける。そのポイントに対応する係数 α が設定される。

40 【0149】上記の5つのポイントに対応する5つの係数 α を、テーブルに記憶させ、係数 α を変更可能にしてもよい。上記の方法により、ボリューム操作子61が一番近いポイントを検出したら、そのポイントに対応する係数 α をテーブルから読み出し、係数 α を決定する。

【0150】係数 α をテーブル化することにより、演算では行いにくいものであるが、操作子位置と係数 α との対応関係を決めることができる。すなわち、テーブルにより、操作子位置と係数 α の対応関係を表す様々なカーブを決めることができる。

【0151】図19は、バランス操作子62による音量バランス制御を説明するための図である。図17ではバランス操作子62を垂直方向に操作可能に示しているが、説明の便宜上、図19ではバランス操作子62を水平方向に操作可能のように示す。

【0152】実線で示すMIDIデータ特性線71は、バランス操作子62の操作位置とMIDIデータの音量係数 β との関係を示す。破線で示すオーディオデータ特性線72は、バランス操作子62の操作位置とオーディオデータの音量係数 β との関係を示す。音量係数 β は、音量の大きさに対応する。

【0153】バランス操作子62を操作可能範囲の中間位置aに移動させると、MIDIデータの音量係数 β が1、オーディオデータの音量係数 β も1に設定される。すなわち、MIDIデータの音量とオーディオデータの音量の相対バランスが等しく設定される。

【0154】バランス操作子62を中間位置aよりも左に移動させると、MIDIデータの音量係数 $\beta=1$ は変化せず、オーディオデータの音量係数 β が小さくなる。すなわち、MIDIデータの音量がオーディオデータの音量よりも相対的に大きくなる。

【0155】逆に、バランス操作子62を中間位置aよりも右に移動させると、オーディオデータの音量係数 $\beta=1$ は変化せず、MIDIデータの音量係数 β が小さくなる。すなわち、オーディオデータの音量がMIDIデータの音量よりも相対的に大きくなる。

【0156】例えば、バランス操作子62を操作位置bに移動させると、オーディオデータの係数 β は1となり、MIDIデータの係数 β は0.6になる。

【0157】次に、MIDIデータの音量バランス制御方法を説明する。実際のMIDIデータのボリュームVtr2は、上記のトラックボリュームVtr1にバランス係数 β を乗じた値とする。

【0158】

$$\begin{aligned} Vtr2 &= Vtr1 \times \beta \\ &= Vtr \times \alpha \times \beta \end{aligned}$$

【0159】次に、オーディオデータの音量バランス制御方法を説明する。実際のオーディオデータのボリュームVau2は、上記のボリュームVau1にバランス係数 β を乗じた値とする。

【0160】

$$\begin{aligned} Vau2 &= Vau1 \times \beta \\ &= Vau \times \alpha \times \beta \end{aligned}$$

【0161】MIDIデータとオーディオデータは、データ生成方法が異なるので、両者の音量バランスがとれていないことがある。バランス操作子62を用いて両者の音量バランスをとることができるので、ユーザは、自己の好みに応じた音量バランスで、MIDIデータとオーディオデータの同時再生を行うことができる。

【0162】バランス係数 β は、ボリューム係数 α の場

合同様に、図18(A)に示す方法で求めてもよいし、図18(B)に示す方法で求めてもよい。

【0163】図20は、係数 α 又は β の第1の決定処理を示すフローチャートである。この処理は、図18

(A)に示した方法に対応する処理である。

【0164】ステップSJ1では、ボリューム操作子61又はバランス操作子62の変化を検出したか否かをチェックする。変化を検出したときにはステップSJ2へ進み、変化を検出しないときには何もせずに処理を終了する。

【0165】ステップSJ2では、ボリューム操作子61又はバランス操作子62の位置から基準点(例えば最下位置)までの距離xを取得する。

【0166】ステップSJ3では、ボリューム操作子61又はバランス操作子62の全範囲の距離yと基準点までの距離xとの比 x/y を取得する。ステップSJ4では、求めた比 x/y を係数 α として設定し、又は求めた比 x/y を基にMIDIデータ及びオーディオデータの係数 β を設定し、記憶する。その後、処理を終了する。

【0167】図21は、係数 α 又は β の第2の決定処理を示すフローチャートである。この処理は、図18

(B)に示した方法に対応する処理である。

【0168】ステップSK1では、ボリューム操作子61又はバランス操作子62の変化を検出したか否かをチェックする。変化を検出したときにはステップSK2へ進み、変化を検出しないときには何もせずに処理を終了する。

【0169】ステップSK2では、ボリューム操作子61又はバランス操作子62の位置を検出する。

【0170】ステップSK3では、予め決められた規定位置(例えば5ポイント)の中で操作子61又は62の最寄りのポイント位置を検出する。すなわち、規定位置から操作子61又は62までの距離が最短のポイント位置を採用する。

【0171】ステップSK4では、採用したポイント位置に規定されている係数 α 又は β をテーブルから読み出し、記憶部(例えば図3のRAM24)に記憶する。その後、処理を終了する。

【0172】図22は、MIDIデータの音量制御を行うための機能ブロック図である。トラックボリュームVtrは、上記のように、受信したコントロールチェンジメッセージに対応する値である。乗算器85は、トラックボリュームVtrにボリューム係数 α を乗じて、新たなトラックボリュームVtr1を生成する。

【0173】 $Vtr1 = Vtr \times \alpha$
乗算器84は、トラックボリュームVtr1にバランス係数 β を乗じて、新たなトラックボリュームVtr2を生成する。

【0174】 $Vtr2 = Vtr1 \times \beta$
音量パラメータPR1は、上記のトラックボリュームと

は別の音量パラメータであり、例えば、ベロシティやエクスペリション等であり、音量演算器81に入力される。音量演算器81は、音量パラメータPR1を基に演算を行い、その演算結果としてパラメータPR2を出力する。

【0175】乗算器83は、トラックボリュームVtr2にパラメータPR2を乗じて、新たなトラックボリュームVtr3を生成する。

【0176】 $Vtr3 = Vtr2 \times PR2$

音源LSI86には、トラックボリュームVtr3及びパラメータPR3が入力される。パラメータPR3は、音量パラメータ以外のパラメータであり、例えば音高情報や音色情報等である。音源LSI86は、トラックボリュームVtr3に応じて楽音信号の音量を制御する。

【0177】音源LSI86は、MIDIチャンネル毎に制御を行うことが可能である。全チャンネルを統一して音量制御してもよいし、MIDIチャンネル毎に個別に音量制御を行ってもよい。

【0178】なお、音源LSI86を他の音源に代えることができる。音源は、専用のハードウェアを用いて構成するものに限らず、DSP+マイクロプログラムを用いて構成してもよいし、CPU+ソフトウェアのプログラム（ソフト音源）で構成するようにしてもよい。

【0179】図23は、ホームコンピュータのRAM24（図3）に記憶されるパラメータ87を示す。

【0180】パラメータ87は、MIDIチャンネル（音源パート）毎に同種類のパラメータを有する。例えば、第1チャンネルのパラメータ87a及び第2チャンネルのパラメータ87b等を有する。

【0181】各チャンネルのパラメータは、トラックボリュームVtr、ボリューム係数 α 、バランス係数 β 、音量パラメータの演算結果PR2、音量パラメータPR1、パラメータPR3を有する。

【0182】1つのボリューム操作子61及び1つのバランス操作子62により全MIDIチャンネルの音量を制御する場合には、全チャンネルのボリューム係数 α 及びバランス係数 β は同じになる。MIDIチャンネル毎に操作子61又は62を設ければ、MIDIチャンネル毎に異なる係数 α 又は β を設定することできる。

【0183】図24は、上記のMIDIデータの音量制御処理を示すフローチャートである。

【0184】音量制御は、2つのイベント発生の中のいずれのイベント発生によっても行われる。第1のイベント発生は、演奏会場からMIDIデータを受信した場合であり、ステップSL1の処理から始まる。第2のイベント発生は、ボリューム操作子61又はバランス操作子62を操作した場合であり、ステップSL10の処理から始まる。

【0185】まず、MIDIデータを受信した場合を説明する。ステップSL1では、再生バッファよりMIDI

Iデータを読み出す。

【0186】ステップSL2では、受信したMIDIデータがトラックボリューム（コントロールチェンジメッセージ）であるか否かをチェックする。トラックボリュームであれば、yesの矢印に従い、ステップSL4へ進み、トラックボリュームでなければ、noの矢印に従い、ステップSL3へ進む。

【0187】ステップSL4では、検出されたトラックボリュームVtrを、対応するMIDIチャンネル（音源パート）別の領域87（図23）に格納する。

【0188】ステップSL5では、決定された係数 α 及び β をパート別領域87から読み出し、トラックボリュームVtrに掛け算し、新たなトラックボリュームVtr2を算出する。

【0189】 $Vtr2 = Vtr \times \alpha \times \beta$

ステップSL6では、パート別領域87から他の音量パラメータPR1を読み出し、それらを基に演算を行い、パラメータPR2を求める。次に、そのパラメータPR2をトラックボリュームVtr2に掛け算し、トラックボリュームVtr3を求める。

【0190】 $Vtr3 = Vtr2 \times PR2$

ステップSL7では、上記のトラックボリュームVtr3を音源LSI86（図22）用のフォーマットに変換し、音源LSI86に書き込む。音源LSI86は、トラックボリュームVtr3に応じて音量を制御する。その後、処理を終了する。

【0191】上記のステップSL3では、受信したMIDIデータが他の音量パラメータPR1であるか否かをチェックする。音量パラメータPR1であれば、yesの矢印に従い、ステップSL8へ進み、音量パラメータPR1でなければ、noの矢印に従い、音量制御をせずに処理を終了する。

【0192】ステップSL8では、パート別領域87から他の音量パラメータPR1を読み出す。すなわち、音量パラメータPR1が複数ある場合において、そのうちの1つのみを受信した場合には、それ以外の音量パラメータPR1をパート別領域87から読み出す。

【0193】次に、全ての音量パラメータPR1を基に演算を行い、演算結果のパラメータPR2を求める。続いて、そのパラメータPR2を当該領域87に格納する。

【0194】ステップSL9では、パート別領域87からトラックボリュームVtr、ボリューム係数 α 、バランス係数 β 、演算結果のパラメータPR2を読み出し、それらを掛け算し、トラックボリュームVTR3を求める。

【0195】 $Vtr3 = Vtr \times \alpha \times \beta \times PR2$

その後、ステップSL7へ進み、上記と同様に、トラックボリュームVtr3をフォーマット変換して音源LSI86に書き込み、処理を終了する。

【0196】次に、ユーザがボリューム操作子61又はバランス操作子62を操作した場合を説明する。

【0197】ステップSL10では、ボリューム操作子61又はバランス操作子62の変化を検出したか否かをチェックする。変化を検出したときには、ステップSL11へ進み、変化を検出しないときには、音量制御を行わずに処理を終了する。

【0198】ステップSL11では、係数決定処理（図20又は図21）によりボリューム係数 α 又はバランス係数 β を決定し、パート別領域87に記憶する。

【0199】ステップSL12では、パート毎に音量制御を行う場合には、パート毎に他の音量パラメータPR1を読み出し、それらの演算を行い、パート毎の演算結果パラメータPR2を求める。全パートを一括して音量制御する場合には、全パート共通の演算結果パラメータPR2を求める。

【0200】次に、パート別領域87からトラックボリュームVtr、ボリューム係数 α 、バランス係数 β を読み出し、それらをパラメータPR2に掛け算し、トラックボリュームVtr3を求める。

【0201】 $Vtr3 = Vtr \times \alpha \times \beta \times PR2$

その後、ステップSL7へ進み、上記と同様に、トラックボリュームVtr3をフォーマット変換して音源LSI86に書き込み、処理を終了する。

【0202】図25は、オーディオデータの音量制御を行うための機能ブロック図である。D/Aコンバータ91には、演奏会場から受信したデジタル形式のオーディオデータが入力される。D/Aコンバータ91は、オーディオデータをデジタル形式からアナログ形式に変換する。フィルタ92は、例えばローパスフィルタであり、アナログ形式のオーディオデータに対してフィルタ処理を行う。

【0203】アンプ93は、ボリューム係数 α 及びバランス係数 β に応じて、フィルタ処理されたオーディオデータに対して音量制御を行う。アンプ93の最大ボリュームをボリュームVauとすると、アンプ93は、オーディオデータを次式によるボリュームVau2に制御して出力する。

【0204】 $Vau2 = Vau \times \alpha \times \beta$

D/Aコンバータ91、フィルタ92及びアンプ93は、コーデック回路27b（図3）で代用することができる。アンプ93の音量制御は、デジタル制御が可能であり、アンプ93のパラメータを指定するだけでよい。

【0205】なお、ホームコンピュータのオペレーティングシステム（OS）としてWindowsを使用する場合には、コントロールパネルで音量指定することができる。この場合は、コントロールパネルによる音量制御と同等の処理により、アンプ93の音量制御を行うようにしてもよい。

【0206】音声出力装置94は、例えばスピーカであ

り、音量制御されたオーディオデータを基に発音する。

【0207】図26は、上記のオーディオデータの音量制御処理を示すフローチャートである。

【0208】ステップSM1では、ボリューム操作子61又はバランス操作子62の変化を検出したか否かをチェックする。変化を検出したときには、ステップSM2へ進み、変化を検出しないときには、音量制御を行わずに処理を終了する。

【0209】ステップSM2では、係数決定処理（図20又は図21）によりボリューム係数 α 又はバランス係数 β を決定し、記憶装置（例えば図3のRAM24）に記憶する。

【0210】次に、記憶装置から係数 α 及び β を読み出し、ボリュームVau2（ $= Vau \times \alpha \times \beta$ ）を求める。そのボリュームVau2をアンプ93（図25）用にフォーマット変換し、アンプ93の制御部に書き込む。アンプ93は、係数 α 及び β に応じて音量制御を行い、音声出力装置94にオーディオデータを発音させる。その後、処理を終了する。

【0211】ユーザは、ボリューム操作子61を操作することにより、MIDIデータ及び／又はオーディオデータの音量を制御することができる。その操作に応じて、ボリューム操作子61の位置を変化させて表示すれば、ユーザは音量の大きさを認識することができる。

【0212】また、ユーザは、バランス操作子62を操作することにより、MIDIデータとオーディオデータとの音量バランスを制御することができる。その際、一方のデータの音量を固定し、他方のデータの音量を小さくすることにより、音量バランスをとることが好ましい。この方法によれば、ユーザは聴覚上音量バランスの変化を確認しやすく、音量バランスの調整が容易になる。

【0213】なお、ボリューム操作子61やバランス操作子62は、音量制御用の操作子であるが、その他の楽音パラメータを制御するための操作子を設けてもよい。その場合は、音量に代えて他の楽音パラメータを制御パラメータとして用いればよい。他の楽音パラメータは、例えば効果パラメータ（イコライザやフィルタ等）である。

【0214】以上実施例に沿って本発明を説明したが、本発明はこれらに制限されるものではない。例えば、種々の変更、改良、組み合わせ等が可能なことは当業者に自明であろう。

【0215】

【発明の効果】以上説明したように、本発明によれば、第1及び第2の楽音情報の各々と、各楽音情報に付与された時間情報とを関連付けて送信することにより、上記の時間情報を基に第1及び第2の楽音情報の同期をとって再生装置に出力することができる。同期をとることにより、第1及び第2の楽音情報は互いにタイミングがあ

って再生される。

【0216】また、指定された楽音パラメータに応じて、MIDIデータ及び音声データを制御して再生を指示することにより、両データの制御を容易かつ適切に行うことができる。

【図面の簡単な説明】

【図1】 楽音情報の通信ネットワークを示す図である。

【図2】 送信端であるエンコーダーと受信端であるホームコンピュータの接続を示す図である。

【図3】 エンコーダー及びホームコンピュータのハードウェアの構成を示す図である。

【図4】 図4(A)はエンコーダーのRAMを示し、図4(B)はホームコンピュータのRAMを示す図である。

【図5】 図5(A)はMIDIデータパケットの構造を示し、図5(B)は音声データパケットの構造を示す図である。

【図6】 音声データのタイミングの調整方法を示すタイミングチャートである。

【図7】 MIDIデータのタイミングの調整方法を示す図である。

【図8】 サーバの第1の処理を示すフローチャートである。

【図9】 サーバの第2の処理を示すフローチャートである。

【図10】 クライアントのパケット受信処理を示すフローチャートである。

【図11】 図10のステップSC8におけるMIDI再生モジュール処理の詳細を示すフローチャートである。

【図12】 割り込み処理を示すフローチャートである。

【図13】 図10のステップSC11における音声再生モジュール処理の詳細を示すフローチャートである。

【図14】 第1のスケジューラ処理を示すフローチャートである。

【図15】 第2のスケジューラ処理を示すフローチャートである。

【図16】 第3のスケジューラ処理を示すフローチャートである。

【図17】 ボリューム操作子及びバランス操作子を含む入力画面を示す図である。

【図18】 ボリューム操作子によるボリュームの制御方法を示す図である。

【図19】 バランス操作子による音量バランスの制御方法を示す図である。

【図20】 音量係数の第1の決定処理を示すフローチャートである。

【図21】 音量係数の第2の決定処理を示すフローチャートである。

【図22】 MIDIデータの音量制御を行うための機能ブロック図である。

【図23】 MIDIチャンネル毎に記憶されるパラメータを示す図である。

【図24】 MIDIデータの音量制御処理を示すフローチャートである。

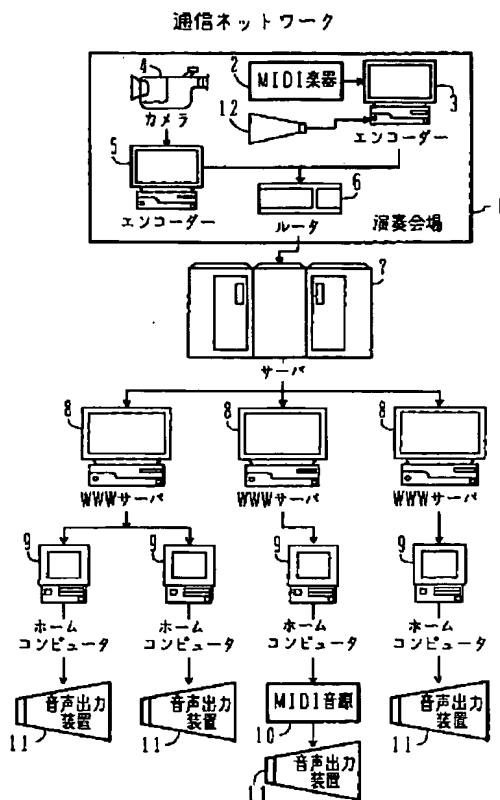
【図25】 オーディオデータ（音声データ）の音量制御を行うための機能ブロック図である。

【図26】 オーディオデータ（音声データ）の音量制御処理を示すフローチャートである。

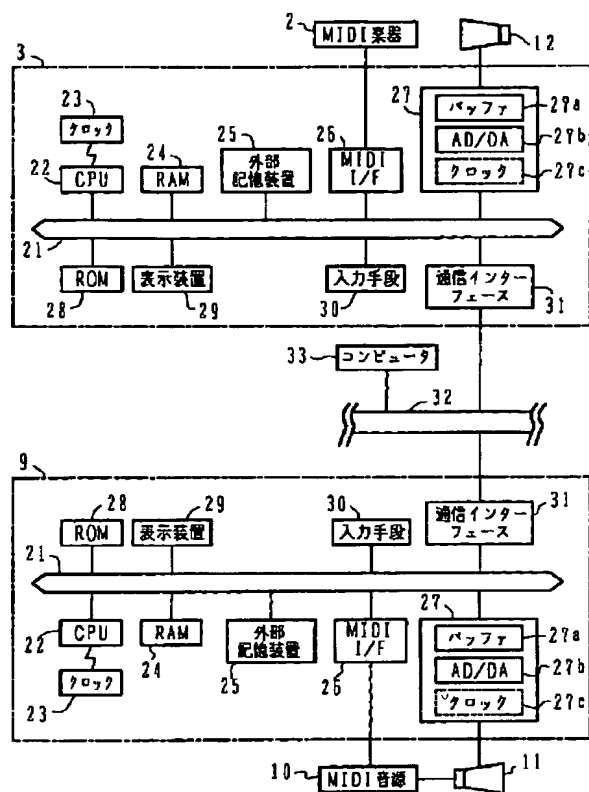
【符号の説明】

1 演奏会場、 2 MIDI楽器、 3, 5 エンコーダー、 4 カメラ、 6 ルータ、 7 サーバ、 8 WWWサーバ、 9 ホームコンピュータ、 10 MIDI音源、 11 音声出力装置、 12 音声入力装置、 21 バス、 22 CPU、 23 システムクロック、 24 RAM、 24 a MIDIデータ用送信バッファ、 24 b 音声データ用送信バッファ、 24 c MIDIデータ用受信バッファ、 24 d 音声データ用受信バッファ、 24 e MIDIデータ用再生バッファ、 24 f 音声データ用再生バッファ、 25 外部記憶装置、 26 MIDIインタフェース、 27 サウンドカード、 27 a バッファ、 27 b コーデック回路、 27 c クロック、 28 ROM、 29 表示装置、 30 入力手段、 31 通信インタフェース、 32 通信ネットワーク、 33 コンピュータ、 41 タイムスタンプ、 42 識別子(ID)、 43 パケットサイズ、 44 MIDIデータ、 45, 47 MIDIイベント、 46 デルタタイム、 48 デジタル音声データ、 49 MIDIデータパケット、 50 音声データパケット、 61 ボリューム操作子、 62 バランス操作子、 63 再生ボタン、 64 停止ボタン、 65 MIDIデータ表示ランプ、 66 オーディオデータ表示ランプ、 67 その他の操作ボタン、 71 MIDIデータ特性線、 72 オーディオデータ特性線、 81 音量演算器、 83, 84, 85 乗算器、 86 音源LSI、 87 パラメータ、 91 D/Aコンバータ、 92 フィルタ、 93 アンプ、 94 音声出力装置

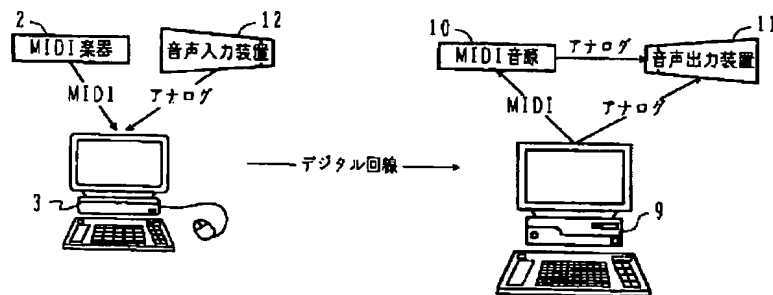
【図1】



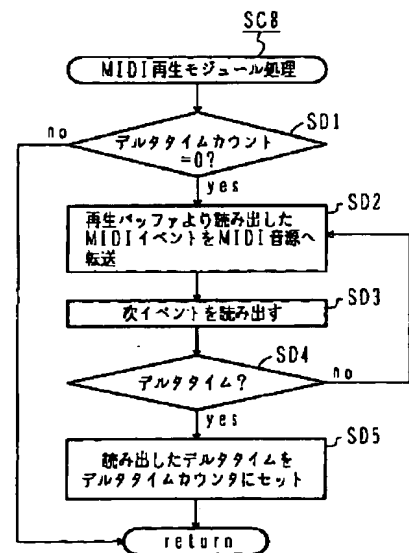
【図3】



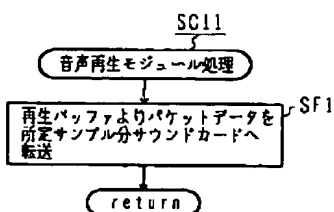
【図2】



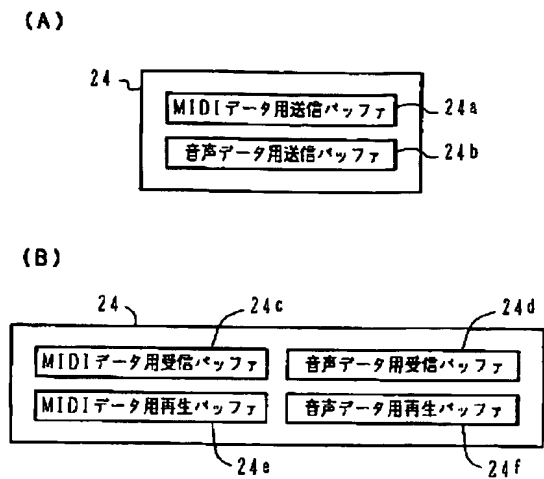
【図11】



【図13】

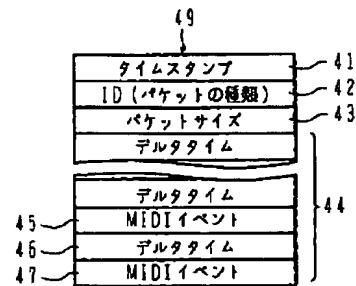


【図4】

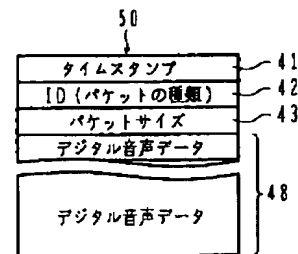


【図5】

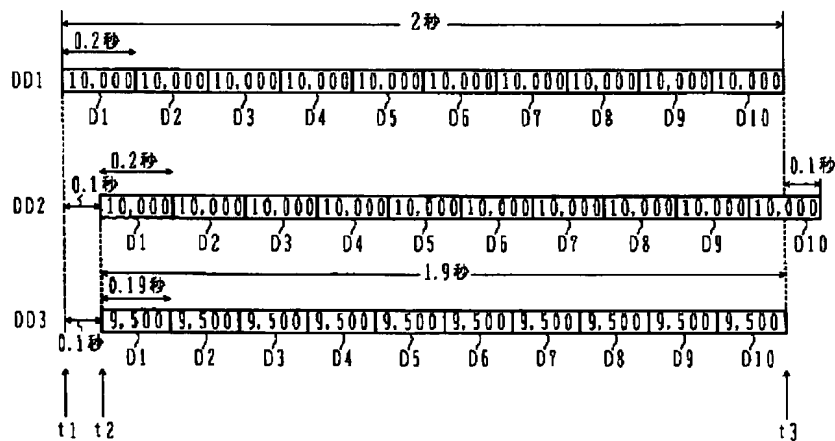
(A) MIDIデータパケット構造



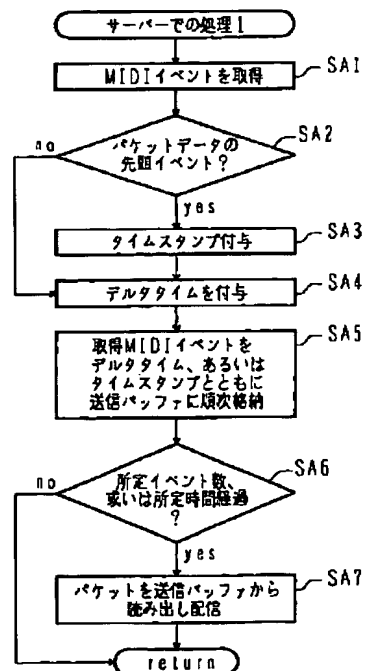
(B) 音声データパケット構造



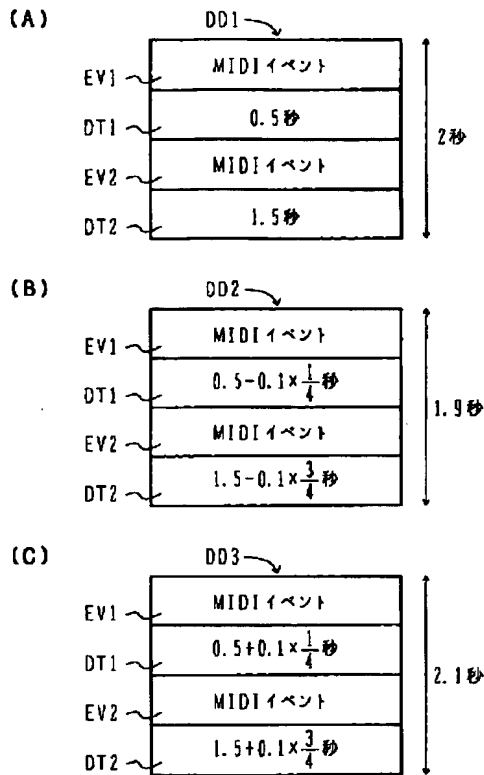
【図6】



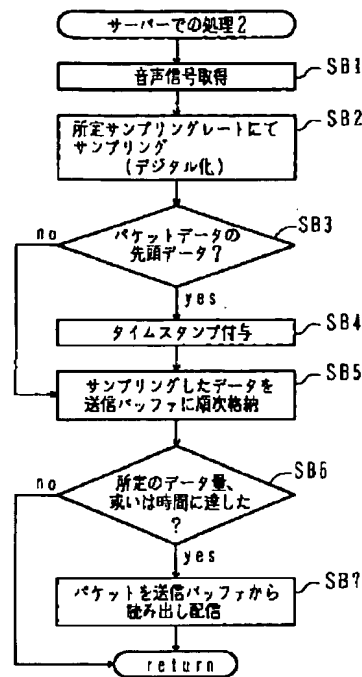
【図8】



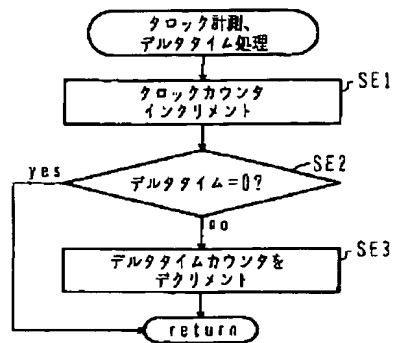
【図7】



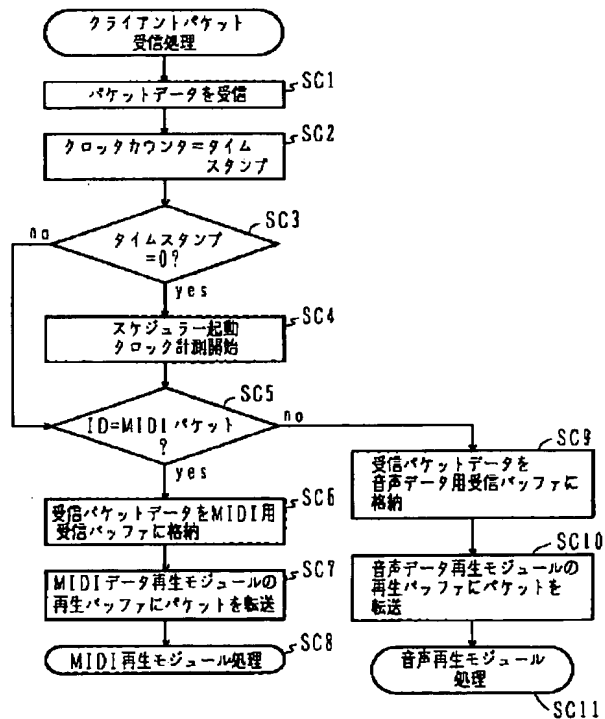
【図9】



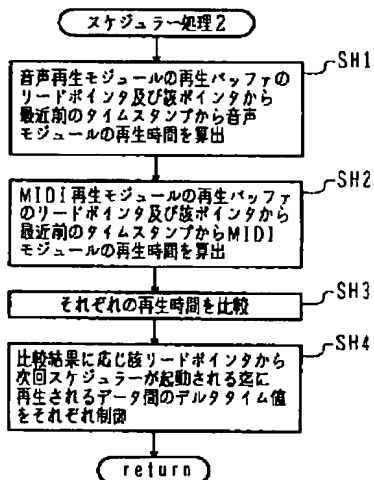
【図12】



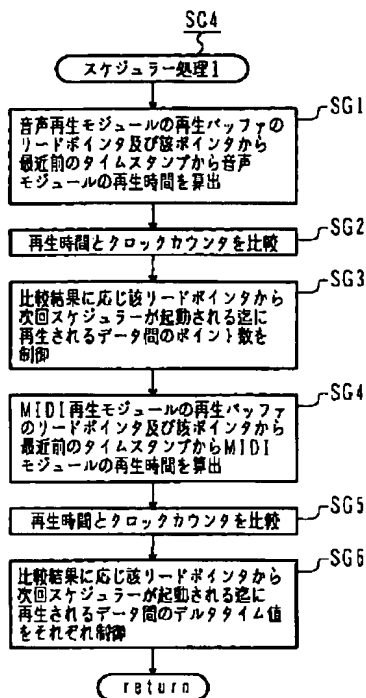
【図10】



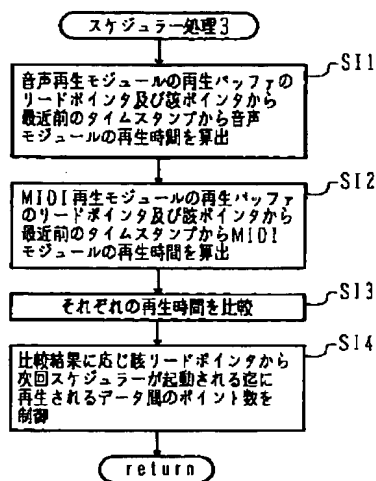
【図15】



【図14】

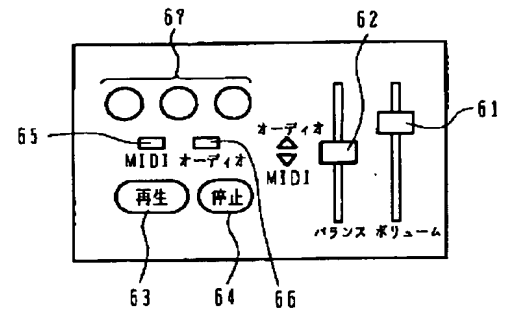


【図16】

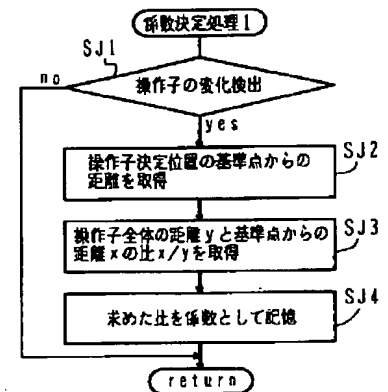


【図17】

入力画面



【図20】

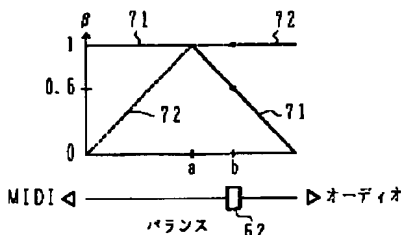


【図19】

バランス操作子

----- : オーディオ

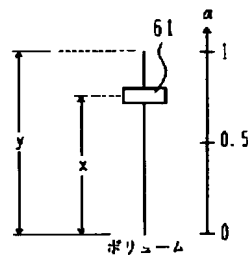
————— : MIDI



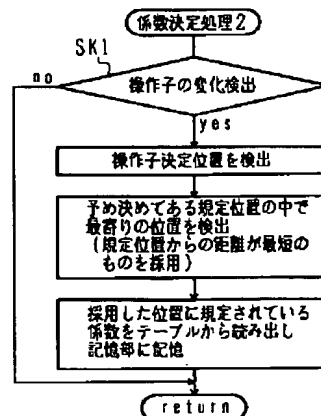
【図18】

ボリューム操作子

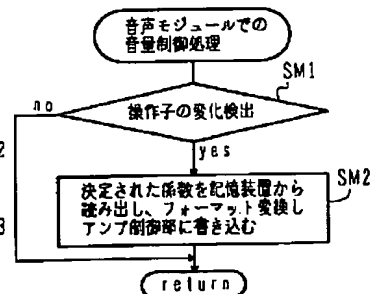
(A)



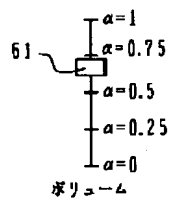
【図21】



【図26】

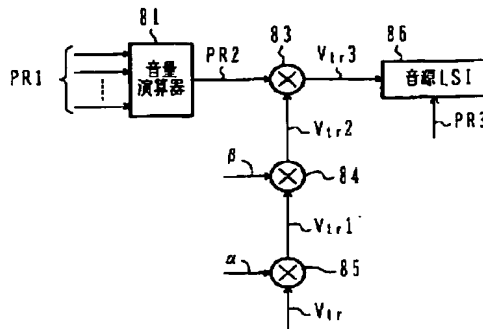


(B)



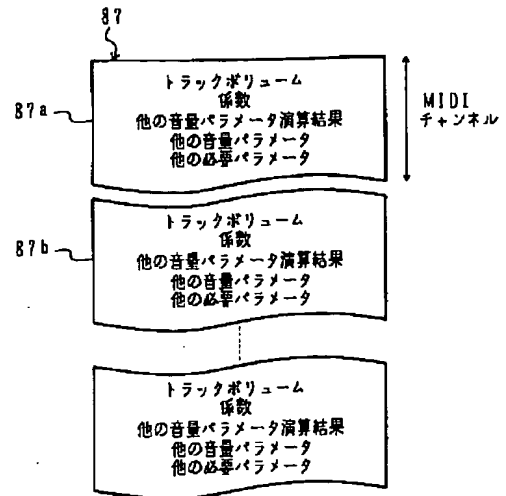
【図22】

MIDI データ 音量制御

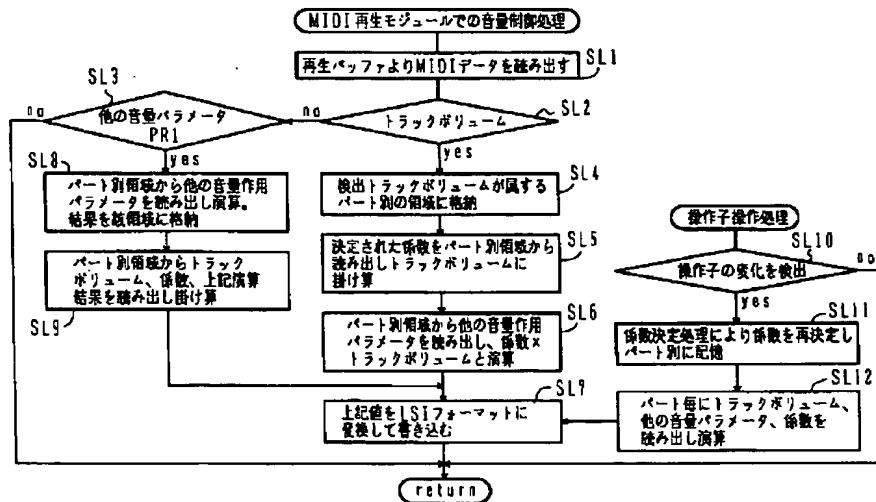


【図23】

メモリマップ



【図24】



【図25】

オーディオデータ 音量制御

